

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО**

*Факультет інформатики та обчислювальної техніки*

(назва факультету, інституту)

*Кафедра автоматизованих систем обробки інформації і управління*

(назва кафедри)

"На правах рукопису"

УДК 004.021:37.012.85

«До захисту допущено»

В.о.завідувача кафедри

О.А.Павлов

(підпис)

(ініціали, прізвище)

“ 02 ” грудня 20 19 р.

**МАГІСТЕРСЬКА ДИСЕРТАЦІЯ**

**на здобуття ступеня магістра**

за спеціальністю 126 Інформаційні системи та технології

(код та назва спеціальності)

ОПП Інформаційні управляючі системи та технології

(код та назва спеціалізації)

на тему: Система підтримки методики навчального процесу, заснованої на  
методі інтервальних повторень

Виконав: студент

VI курсу групи ІС-81мп

(шифр групи)

Романчук Іван Олександрович

(прізвище, ім'я, по батькові)

(підпис)

Науковий керівник

доц., к.т.н. Попенко В.Д.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Консультант

проф., д.т.н. Томашевський В.М.

(науковий ступінь, вчене звання, прізвище, ініціали)

(підпис)

Рецензент

доц., к.т.н., доц. Пасько В.П.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Засвідчую, що у цій магістерській дисертації  
немає запозичень з праць інших авторів без  
відповідних посилань.

Студент

(підпис)

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім. ІГОРЯ СІКОРСЬКОГО»**

Факультет інформатики та обчислювальної техніки  
(повна назва)

Кафедра автоматизованих систем обробки інформації та управління  
(повна назва)

Рівень вищої освіти другий (магістерський) за освітньо-професійною програмою

Спеціальність 126 Інформаційні системи та технології  
(код і назва)

ОПП Інформаційні управляючі системи та технології  
(код і назва)

ЗАТВЕРДЖУЮ  
В.о.завідувача кафедри  
О.А.Павлов  
(підпис) (ініціали, прізвище)  
« 02 » вересня 2019 р.

**ЗАВДАННЯ**

**на магістерську дисертацію студенту  
Романчуку Івану Олександровичу**

(прізвище, ім'я, по батькові)

1. Тема дисертації Система підтримки методики навчального процесу,  
заснованої на методі інтервальних повторень

науковий керівник дисертації Попенко Володимир Дмитрович, к.т.н.  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від “ 28 ” жовтня 20 19 р. № 3770-с

2. Строк подання студентом дисертації “ 02 ” грудня 20 19 р.

3. Об'єкт дослідження Процес підтримки методики навчального процесу

4. Перелік завдань, які потрібно розробити 1) виконати огляд відомих результатів з розв'язання поставленої в рамках роботи задачі; 2) знайти оптимальний термін самостійного тестування з урахуванням обмежень навчального процесу; 3) розробити програмну реалізацію розробленого алгоритму планування набору питань самостійного тестування; 4) виконати аналіз отриманих результатів.

5. Орієнтовний перелік ілюстративного матеріалу 1) Схема структурна діяльності до оптимізації; 2) Схема структурна діяльності після оптимізації; 3) Модель «сутність-зв'язок»; 4) Крива забування в разі невдалого проходження тесту; 5) Крива забування в разі вдалого проходження тесту; 6) Місячна ймовірність запам'ятовування, внески випадків невдалого (червона крива) і вдалого (синя крива) тестування; 7) Місячна ймовірність запам'ятовування, сумарний графік; 8) Результати проведеного експерименту з використанням методу інтервальних повторень.

6. Орієнтовний перелік публікацій Дві публікації: одна стаття у фаховому виданні, одні тези доповіді на науковій конференції.

#### 7. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

8. Дата видачі завдання “ 2 ” вересня 20 19 р.

#### Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Строк виконання етапів магістерської дисертації	Примітка
1	Систематизація результатів огляду літератури	05.09	
2	Порівняльний аналіз існуючих методів розв'язання задачі	12.09	
3	Постановка та формалізація математичної моделі задачі	17.09	
4	Модифікація існуючих методів розв'язання задачі	25.09	
5	Розробка інформаційного та програмного забезпечення	02.10	
7	Проведення експериментальних досліджень розроблених алгоритмів	18.10	
8	Оформлення документації	01.11	
9	Подання роботи на попередній захист	20.11	
10	Подання роботи на основний захист	02.12	

Студент

\_\_\_\_\_  
(підпис)

*І.О. Романчук*

\_\_\_\_\_  
(ініціали, прізвище)

Науковий керівник

\_\_\_\_\_  
(підпис)

*В.Д. Попенко*

\_\_\_\_\_  
(ініціали, прізвище)

## РЕФЕРАТ

Магістерська дисертація: 103 с., 26 рис., 27 табл., 47 джерел, 1 додаток.

**Актуальність.** Людство постійно потребує навчання. Кожна людина ніколи не обмежується школою, коледжем або вищим навчальним закладом. Після отримання диплому теж потрібно постійно вивчати щось нове, тренувати навички, розвивати себе. Адже життя не стоїть на місці. Тому і постало питання у тому, як можна покращити цей процес, як працює людський мозок і запам'ятовування, чи можливо математично вирахувати якусь залежність і покращити ці результати, не керуючись інтуїтивними передбаченнями чи припущеннями. Саме цьому присвячена дана робота, в якій розглянуто питання навчання, покращення існуючих методик у питаннях неабстрактного вивчення нового матеріалу, а саме як це можна застосувати на практиці. Результати дослідження будуть корисні як і для учнів, студентів, так і для викладачів, які зможуть використовувати або покращувати методику викладання матеріалу, а також покращити вивчення та запам'ятовування нового. В Україні поки що не існує такої системи чи сервісу, які б могли вирішувати цю задачу. Так, вже стали з'являтися певні онлайн-курси, сервіси з тренування певних навичок, але вони не покривають описану область. Тому, ця робота є актуальною на сьогоднішній день і вносить певну новизну для системи освіти в Україні і світі.

У зв'язку з цим актуальною є розробка програмного продукту для підтримки методики навчального процесу, заснованої на методі інтервальних повторень, що дозволить полегшити процес навчання та покращити запам'ятовуваність матеріалу.

**Зв'язок роботи з науковими програмами, планами, темами.** Робота виконувалась на кафедрі автоматизованих систем обробки інформації та управління Національного технічного університету України «Київський політехнічний інститут ім. Ігоря Сікорського» в рамках теми «Методи та технології в задачах пошуку та збереження даних» (№ ДР0117U000915).

**Мета дослідження** – підвищення ефективності підтримки методики навчання та його планування за рахунок використання тестування в рамках обґрунтованої методики.

Для досягнення мети необхідно виконати наступні **завдання**:

- виконати огляд відомих результатів з розв’язання поставленої в рамках роботи задачі;
- знайти оптимальний термін самостійного тестування з урахуванням обмежень навчального процесу;
- розробити програмну реалізацію розробленого алгоритму планування набору питань самостійного тестування;
- виконати аналіз отриманих результатів.

**Об’єкт дослідження** – процес підтримки методики та планування навчання з урахуванням індивідуальних особливостей учнів.

**Предмет дослідження** – методи підтримки методики та планування навчання з урахуванням індивідуальних особливостей студентів або учнів.

**Наукова новизна отриманих результатів** полягає у розроблених підходах та методах вирішення поставленої задачі «Визначення оптимального терміну самостійного тестування з урахуванням обмежень навчального процесу» з використанням поняття інтервальних повторень та кривої забування Еббінгауза. Використаний створений алгоритм на основі вирішеної задачі пошуку терміну проведення самостійного тестування дозволяє істотно підвищити запам’ятовуваність матеріалу під час навчання.

**Публікації.** Матеріали роботи опубліковані в тезах науково-практичної конференції молодих вчених та студентів «Інформаційні системи та технології управління» (ІСТУ-2019), стаття прийнята до публікації у фаховому виданні "Наукові записки. Серія: Педагогічні науки" ЦДПУ ім. Володимира Винниченка.

**МЕТОДИКИ НАВЧАННЯ, СИСТЕМА ПІДТРИМКИ, СИСТЕМА ДИСТАНЦІЙНОГО НАВЧАННЯ, ПОШУК ТЕРМІНУ ТЕСТУВАННЯ, ЗАПАМ’ЯТОВУВАНІСТЬ.**

## ABSTRACT

Master's thesis: 103 pp., 26 images, 27 tables, 47 sources, 1 supplement. **Topicality.**

Humanity is constantly in need of training. Everyone is never limited to a school, college, or university. After receiving a diploma, you also need to constantly learn something new, train your skills, and develop yourself. After all, life does not stand still. Therefore, the question is about how this process can be improved, how the human brain and memory work, is it possible to mathematically calculate some kind of dependence and improve these results, guided by intuitive predictions or assumptions. This work is dedicated to this, in which the questions of training, the improvement of existing methods in matters of non-abstract study of new material, are examined, namely how this can be applied in practice. The results of study will be useful both for pupils and students, and for tutors who can use or improve the methodology of teaching material, as well as improve learning and remembering new things. In Ukraine, so far there is no such system or service that could solve this problem. So, certain online courses, services from training certain skills have already begun to appear, but they do not cover the described area. Therefore, this work is relevant today and introduces a certain novelty for the education system in Ukraine and the world.

In this regard, it is urgent to develop a software product to support the methodology of the educational process, based on the spaced repetition methods, which will facilitate the learning process and improve memorability of the material.

**Relationship of work with scientific programs, plans, themes.** The work was carried out at the Department of Automated Systems for Information Processing and Management of the National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute" within the terms of the theme "Methods and technologies in the searching and storing data problems" (state registration number 0117U000915).

**The purpose of the study** is to increase the effectiveness of support for teaching methods and their planning through the use of testing within the terms of a reasonable methodology.

To achieve the goal, the following **tasks** should be done:

- perform a review of the known results for solving the problem posed in the terms of the work;
- find the optimal time for self-testing, taking into account the limitations of the educational process;
- to develop a software implementation of the developed algorithm for planning a set of self-testing questions;
- perform an analysis of the results.

**The object of the research** is the process of supporting the educational methods based on spaced repetition learning, and training planning, taking into account the individual characteristics of students.

**The subject of the research** is support methods and training planning, taking into account the individual characteristics of students or pupils.

**The scientific novelty of the obtained results** consists in the developed approaches and methods for solving the problem “Determining the optimal term for self-testing, taking into account the limitations of the educational process” using the concept of spaced repetition learning and the Ebbinghaus forget curve. The created algorithm is used on the basis of the solved problem of finding the term for self-testing, which can significantly increase the memorability of the material during training.

**Publications.** The materials of the work were published in the theses of the scientific-practical conference of young scientists and students “Information Systems and Management Technologies” 2019, the article was accepted for contribution in the scientific publication “Scientific Notes. Series: Pedagogical Sciences” Volodymyr Vynnychenko Central Ukrainian State Pedagogical University.

TRAINING METHODS, SUPPORT SYSTEM, LEARNING MANAGEMENT SYSTEM, SEARCH OF SELF-TESTING TERMS, MEMORABILITY.

## ЗМІСТ

ВСТУП .....	9
1 ПРОЕКТНІ РІШЕННЯ З РОЗРОБКИ СИСТЕМИ ПІДТРИМКИ МЕТОДИКИ НАВЧАЛЬНОГО ПРОЦЕСУ, ЗАСНОВАНОЇ НА МЕТОДІ ІНТЕРВАЛЬНИХ ПОВТОРЕНЬ .....	10
1.1 Опис бізнес-процесів.....	10
1.1.1 Опис процесу діяльності.....	11
1.1.2 Актори і функції .....	12
1.1.3 Структура бізнес-процесів.....	16
1.2 Огляд аналогів системи, що розробляється .....	17
1.3 Рішення з інформаційного забезпечення .....	25
1.4 Цілі та завдання системи .....	26
Висновок до розділу.....	27
2 МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ ПРОЦЕСІВ ТЕСТУВАННЯ В НАВЧАННІ .....	28
2.1 Огляд наявного навчального процесу («як є») .....	28
2.2 Огляд задач оптимізації навчання і методів їх розв’язання іншими дослідниками .....	29
2.2.1 Огляд досліджень технік запам’ятовування .....	29
2.2.2 Застосування технік запам’ятовування у програмних продуктах .....	33
2.2.3 Метод регресії періоду напівзабування.....	36
2.3 Удосконалення навчального процесу («як має бути») .....	39
2.4 Пошук оптимального терміну самостійного тестування та визначення періоду напівзабування.....	41
2.4.1 Задача пошуку оптимального терміну самостійного тестування .....	41
2.4.2 Визначення періоду напівзабування.....	43
2.5 Дослідження числових параметрів процесу запам’ятовування.....	44
Висновок до розділу.....	46
3 ОПИС ПРОГРАМНОГО ТА ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ .....	48
3.1 Засоби розробки.....	48
3.1.1 Вибір платформи для системи.....	48
3.1.2 Вибір мови програмування.....	51
3.1.3 Вибір фреймворка для розробки веб-застосування.....	51
3.1.4 Вибір середовища розробки .....	56
3.1.5 Вибір сервісу для реалізації backend-складової системи.....	60



	8
3.2 Вимоги до технічного забезпечення.....	62
3.3 Архітектура програмного забезпечення.....	63
3.3.1 Діаграма послідовності .....	63
3.3.2 Діаграма компонентів .....	64
3.4 Настанова користувача .....	65
3.5 Випробування програмного продукту.....	67
3.5.1 Загальні положення .....	68
3.5.2 Результати випробувань.....	68
Висновок до розділу .....	70
4 РОЗРОБКА СТАРТАП-ПРОЕКТУ .....	71
4.1 Опис ідеї проекту .....	71
4.2 Опис конкурентів .....	71
4.3 Технологічний аудит ідеї проекту .....	74
4.4 Аналіз ринкових можливостей запуску стартап-проекту .....	75
4.5 Розроблення ринкової стратегії проекту.....	82
4.6 Розроблення маркетингової програми стартап-проекту.....	84
Висновок до розділу.....	88
ПЕРЕЛІК ПОСИЛАНЬ .....	90
ДОДАТОК А.....	94
Графічний матеріал .....	94
Схема структурна діяльності до оптимізації .....	95
Схема структурна діяльності після автоматизації .....	96
Модель «сутність-зв'язок» .....	97
Крива забування в разі невдалого проходження тесту .....	98
Крива забування в разі вдалого проходження тесту .....	99
Ймовірність запам'ятовування за місяць, внески випадків невдалого (червона крива) і вдалого (синя крива) тестування .....	100
Ймовірність запам'ятовування за місяць, сумарний графік.....	101
Результати проведеного експерименту з використанням методу інтервальних повторень.....	102

## ВСТУП

Сьогодні поняття самостійного навчання є досить актуальним. Існує багато різних сервісів, що пропонують онлайн-курси, онлайн-тренінги, розвитку навичок за допомогою інформаційних систем та технологій. І цей напрям стає дуже популярним. Використовуючи різноманітні методи та сервіси люди намагаються або знайти та опанувати нові навички, або ж розвивати вже наявні в них. Якщо поглянути на історію людства, то людина ніколи не припиняла процес свого розвитку та навчання. І зараз, у XXI сторіччі, коли інформація та технології знаходяться у відкритому доступі, з'являється попит на нові інструменти для навчання, що стимулює появи пропозиції. Також достатньо поширеними є системи дистанційного навчання, які використовуються у всьому світі, в тому числі, у навчальних закладах, що допомагає полегшити певні процеси у навчанні.

Але існує дуже мала кількість систем, які при цьому враховують індивідуальні навички студентів, і які формують індивідуальні навчальні плани для них.

Тому я вирішив створити власну систему підтримки планування та методики навчання, яку можна реально застосувати в освітній галузі у нашій країні для вивчення будь-якої дисципліни, з урахуванням індивідуальних особливостей студентів на основі опрацювання статистичних даних.

Магістерська дисертація присвячена розробці системи підтримки методики навчання, заснованої на методі інтервальних повторень. Мета магістерської дисертації – створення умов для поліпшення процесу навчання шляхом підвищення запам'ятовуваності навчального матеріалу.

**Практичне значення одержаних результатів.** Розроблено систему дистанційного адаптивного навчання, що створює умови для планування навчання та підтримки методики навчання, яку можна застосувати у будь-якому навчальному закладі.

# **1 ПРОЕКТНІ РІШЕННЯ З РОЗРОБКИ СИСТЕМИ ПІДТРИМКИ МЕТОДИКИ НАВЧАЛЬНОГО ПРОЦЕСУ, ЗАСНОВАНОЇ НА МЕТОДІ ІНТЕРВАЛЬНИХ ПОВТОРЕНЬ**

## **1.1 Опис бізнес-процесів**

Сучасний розвиток суспільства визначається стрімким збільшенням швидкості старіння знань, у зв'язку з чим зростають потреби в їх оновленні. Досягнення інформаційних технологій дозволяють вирішити зазначену проблему шляхом широкого застосування у сфері освіти дистанційного навчання. Проте, для забезпечення навчального процесу з цією формою навчання на належному рівні необхідне масштабне застосування потужних комп'ютерних систем, які будуть у змозі забезпечити віддалений (дистанційний) доступ до захищених інформаційних систем і ресурсів навчального призначення таких як віртуальні центри знань, електронні бібліотеки, курси дистанційного навчання, електронні підручники, віртуальні лабораторні практикуми, системи тестування, відеоконференції тощо, та інтелектуальне керування цим доступом [47].

Серед актуальних завдань, пов'язаних з використанням систем дистанційного навчання, особливу значимість набувають web-системи адаптивного навчання, що адаптуються до знань і здібностей студента, адже такі системи мають низку переваг [45]:

- дозволяють зменшити непродуктивні витрати праці викладача;
- стимулюють навчання самих студентів, що змінює провідну роль викладача;
- передбачають диференційний підхід до студентів, який ґрунтується на визнанні того факту, що у різних студентів попередній досвід і рівень знань в одній області різні (кожен студент приходить до процесу оволодіння новими знаннями зі своїм власним інтелектуальним багажем, який і визначає ступінь розуміння ним нового матеріалу і його інтерпретацію, тобто здійснюється переорієнтація від оволодіння всіма студентами одного і того ж матеріалу до оволодіння різними студентами різного матеріалу);
- зводять на новий рівень операції контролю і оцінки результатів навчання;
- гарантують безперервний зв'язок у відносинах «викладач – студент»;

- сприяють індивідуалізації навчальної діяльності (диференціація темпу навчання, складності навчальних завдань тощо);
- підвищують мотивацію отримання знань за рахунок індивідуального підходу;
- сприяють розвитку в студентів продуктивних, творчих функцій мислення, зростанню інтелектуальних здібностей, формуванню операційного стилю мислення [45].

Система, яка розглядається у даній роботі, присвячена підтримці методики навчання, заснованій на методі інтервальних повторень. Вона полягає у плануванні навчання на основі вищезгаданого методу з урахуванням індивідуальних особливостей студентів. Системи дистанційного навчання не зупиняють процес власного розвитку. Але для нашої держави – це своєрідне нововведення, український стартап, який дозволить не просто розвивати запам'ятовуваність матеріалу та тренувати навички користувачам, а ще й реально застосувати в системі освіти, наприклад, у вищих навчальних закладах. Суть створення системи – навчити людей запам'ятовувати матеріал, для неї створений спеціальний алгоритм, який дозволить на основі успішності користувачів (статистики) формувати для них індивідуальні навчальні плани, що створює комфортні умови для студента. Майбутнє – за індивідуальним навчанням, а інформаційні технології дозволяють знайти та реалізувати більш просте рішення усіх цих потреб [1].

### **1.1.1 Опис процесу діяльності**

Запропонована система є похідною від дистанційних систем дистанційного навчання. Особливістю системи, що розглядається, є те, що вона буде визначати оптимальний термін самостійного тестування, планувати набір питань у тестах – вибіркові питання за останнім пройденим матеріалом та повторні питання за слабкими місцями, які були виявлені під час попередніх тестувань.

Система базується на методиці самостійного тестування та варіанті її оптимізації. Наразі процес виглядає наступним чином: існує декілька вкладених циклів навчання: найменший – тижневий, далі модульний, який включає 7-8 попередніх тижневих циклів, а останній – семестровий цикл, який включає 2 модульні

цикли. У процесі діяльності задіяні викладач і користувач, а також система дистанційного навчання, якій і приділена увага у даній роботі. Отже, у процесі діяльності ключовою є саме ця система.

За архітектурою система буде реалізована у вигляді веб-застосування. Це дозволить зберегти універсальність та доступність системи як зі стаціонарних комп'ютерів, так і з мобільних пристроїв, які мають доступ до мережі Інтернет.

### 1.1.2 Актори і функції

Всередині системи можна виділити наступних акторів, які взаємодіють один між одним:

- студент;
- викладач;
- адміністратор.

*Студент* – це особа, що навчається. Може проходити тести з дисциплін, також має доступ до навчальних матеріалів, а ще може комунікувати з іншими студентами, а також викладачами через чати. Функціонал доступний для тих, хто має зареєстрований обліковий запис у системі.

*Викладач* – це особа, що навчає. Викладач має більший набір дій: створення та редагування навчальних курсів, створення та редагування навчальних матеріалів, створення, редагування та налаштування тестів, перегляд результатів тестувань.

*Адміністратор* – особа, яка здійснює управління обліковими записами та налаштовує систему

В таблиці 1.1 нижче представлені основні функції для кожного з акторів.

Таблиця 1.1 – Опис акторів та функцій системи

Актор	Функція	Опис функції
Студент	Реєстрація та авторизація в системі	Студент повинен зареєструватися або авторизуватися для отримання доступу до системи та її функцій

## Продовження таблиці 1.1

Актор	Функція	Опис функції
Студент	Проходження тестів з обраної дисципліни	Студент має змогу пройти доступний тест за дисципліною. Також він має змогу пройти повторний тест, коли система про це його сповістить. Уся статистика збирається системою
	Робота з навчальними матеріалами	Студент має доступ до навчальних матеріалів за дисциплінами, які розміщують викладачі. Також він має можливість завантажити власні матеріали для передачі їх викладачу на перевірку.
	Комунікація через чат	Студент також може комунікувати за допомогою чатів, наявних у системі
	Перегляд статистики проходження тестів і оцінок	Студент має змогу після кожного тестування переглянути статистику правильних і неправильних відповідей на питання. Також студент може за будь-якою дисципліною подивитися історію навчання у вигляді оцінок за тестування, а також побачити проблемні місця.
Викладач	Створення та редагування навчальних курсів	Викладач може створювати та/або редагувати навчальні курси для своїх студентів для обраної дисципліни.
	Створення та редагування навчальних матеріалів	Викладач може створювати та/або редагувати навчальні матеріали, які будуть доступні для студентів за дисципліною

Актор	Функція	Опис функції
Викладач	Створення, редагування, налаштування навчальних тестів	Викладач може створювати або редагувати, а також налаштовувати навчальні тести, які будуть використовуватися у системі для навчання
	Перегляд статистики проходження тестів	Викладач може переглядати статистику, представлену в адмін-центрі. Інформація наявна як для кожного окремого користувача, так і загальна статистика для всіх користувачів
Адміністратор	Управління обліковими записами	Адміністратор здійснює управління обліковими записами користувачів – студентів і викладачів
	Налаштування системи	Адміністратор має повноваження для виконання налаштувань системи

Відповідно до визначених варіантів використання можна побудувати модель варіантів використання.

Система, що розглядається, є похідною від систем дистанційного навчання або дистанційних адаптивних навчальних систем. «Дистанційна адаптивна навчальна система – це складна, багатокомпонентна система, що має тенденцію до безперервних змін і динамічного розвитку. Ефективна робота такої системи не може спиратися лише на емпіричний досвід і інтуїцію розробників. Необхідно застосування системного підходу до адаптивного дистанційного навчання з врахуванням залежності складових системи – підсистем і компонентів» [46].

Функціональна модель призначена для вивчення особливостей роботи (функціонування) системи та її призначення у взаємозв'язку з внутрішніми і зовнішніми елементами. Вона поділяє систему на компоненти за функціональними ознаками. Наприклад, теоретичне навчання, тестування, спілкування тощо. Більшість

пакетів дистанційного навчання розробляються на основі функціональної моделі, яка також використовується у дистанційній адаптивній навчальній системі. Вона зображена на рисунку 1.1.

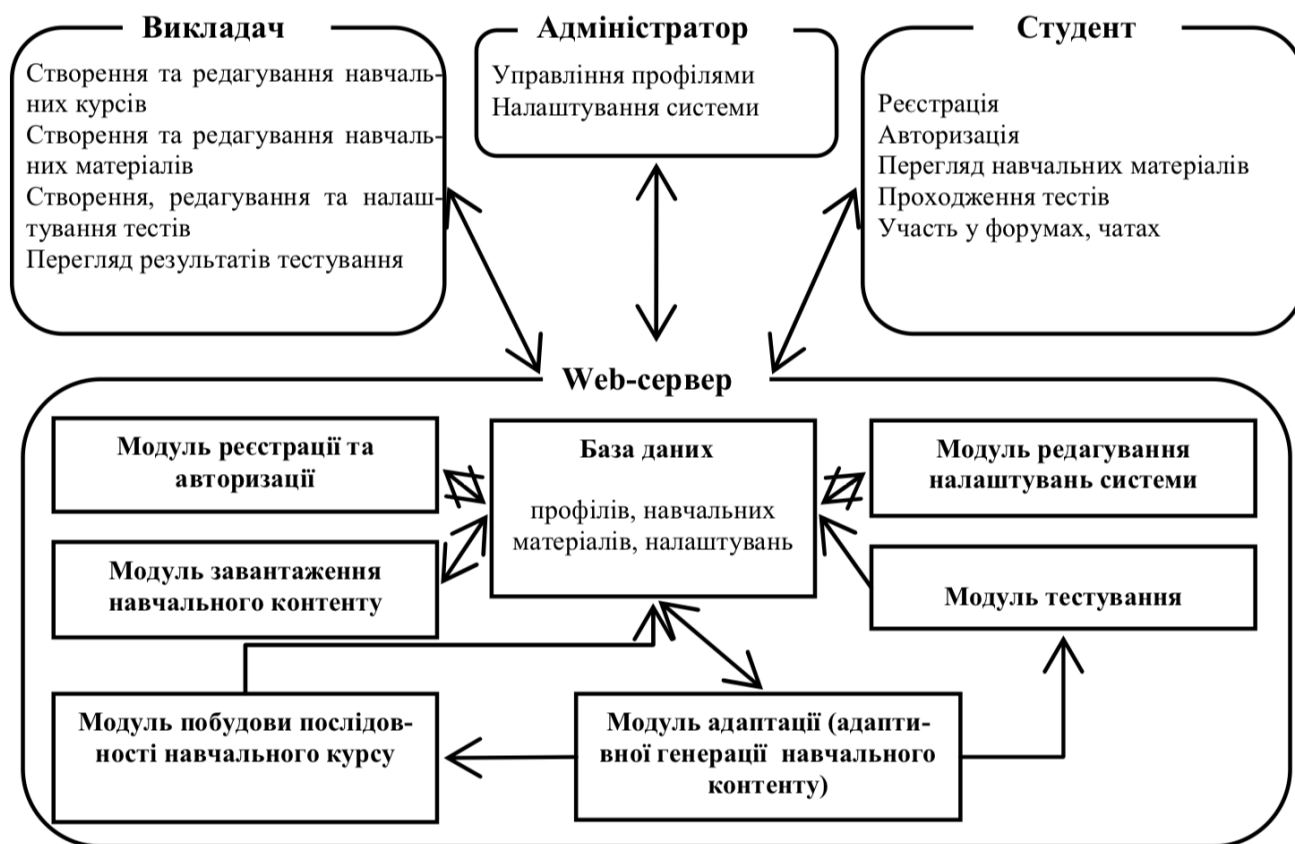


Рисунок 1.1 – Функціональна модель дистанційної адаптивної навчальної системи

Завдяки даній функціональній моделі можна виділити компоненти системи:

- база даних;
- модуль реєстрації та авторизації;
- модуль адміністрування системи (редагування налаштувань системи);
- модуль для роботи із навчальним контентом (завантаження матеріалів);
- модуль тестування (проведення тестування);
- модуль планування тестів (адаптивне генерування тестів).

Визначивши компоненти системи, можна перейти до огляду структури бізнес-процесів.



### 1.1.3 Структура бізнес-процесів

Наразі система має достатньо обмежену роль і відповідні їй функції. Мова не йде про стандартний функціонал – реєстрація, авторизація, адміністрування тощо. Функціонал розглядається відносно безпосередньої взаємодії з користувачем. Варто зазначити, що у моєму навчальному закладі наразі використовується інша, але досить поширена система дистанційного навчання (СДН) – Moodle. І використовується вона як інструмент формування тестів із вибраних викладачем категорій та оцінювання результатів проходження даних тестів студентами. Варто зазначити, що на формування тестів безпосередньо впливає викладач, адже саме він вибирає категорії питань для обраного тесту, визначає, коли певний тест доступний для студентів, а також обмеження по часу та кількості спроб. Отже, основна діяльність системи полягає саме в оцінюванні пройдених студентами тестів та забезпечення саме цього. Схема структурна діяльності до оптимізації наведена у графічному матеріалі, на рисунку Додатка А.1.

Після ж оптимізації система буде трохи інакше брати участь у формуванні тестів. Вона буде враховувати «слабкі місця» з попередніх тестів і додавати їх у кінці до нових тестів, які будуть проходити студенти. Якщо ж матеріал пройдено правильно, то термін повторення буде збільшено. Всі ці питання докладно розглянуті у 2 розділі даної роботи, де описана задача визначення оптимального терміну самостійного тестування. У результаті, в процесі діяльності змінюється трохи тижневий цикл, а саме – студенту додається проходження повторних питань після актуального тесту, що дає змогу відновити в пам'яті пройдений матеріал або запам'ятати його.

Розглянута техніка заснована на методі інтервальних повторень, яка відіграє ключову роль у даній роботі і теж розглянута у розділі. Схема структурна діяльності після оптимізації наведена у графічному матеріалі, на рисунку Додатка А.2.

Для правильного функціонування необхідний збір даних щодо проходження тестів студентами. Система буде записувати результати за кожним питанням, щоб виявляти «слабкі місця» і потім давати ці питання на повторення, щоб у кінцевому результаті підвищити утримання матеріалу у пам'яті людини. Дана статистика буде

не тільки використовуватися системою для формування навчального плану для студента, але вона також буде корисна і для викладача, оскільки він зможе переглянути цю інформацію як у вигляді цифрових даних, так і графічно, при чому як для одного конкретного студента, так і для групи студентів, що дозволить йому також зробити висновок про складні питання в його матеріалі, які потребують додаткової уваги (докладного пояснення на лекції, практичних прикладів тощо) під час вивчення цього матеріалу.

## **1.2 Огляд аналогів системи, що розробляється**

Існує досить велика кількість схожих сервісів. Це пояснюється тенденцією користування технологіями і розвитком систем дистанційного навчання: платформ тренування навичок, онлайн-курсів тощо. Найближчих аналогів не існує, оскільки дана система покриває навчання у загальному, без прив'язки до певної дисципліни, а також вона враховує індивідуальні властивості студентів. Тому, будуть розглянуті «часткові» аналоги для моєї системи.

*Codecademy* [3] – сервіс, який пов'язаний з програмуванням, в якому теж матеріал категоризується за тематикою – вивчення окремої мови, курс створення веб-сайтів, бази даних тощо. Сервіс дає як теоретичний, так і практичний матеріал, дозволяючи одразу застосовувати нові знання на практиці. Із недоліків – один план навчання для всіх (не враховуються індивідуальні особливості студентів), система робить висновок про миттєве запам'ятовування, а не довгострокове; є платні курси. На рисунку 1.2 продемонстровано панель керування курсами у системі, а також розміщення основних елементів користувацького інтерфейсу. Як видно з ілюстрації нижче, інтерфейс доволі простий, не нагромаджений зайвою інформацією. Також на рисунку можна побачити, що сервіс пропонує як безкоштовні курси, так і ті, які доступні для підписників на переплату під назвою «Pro».

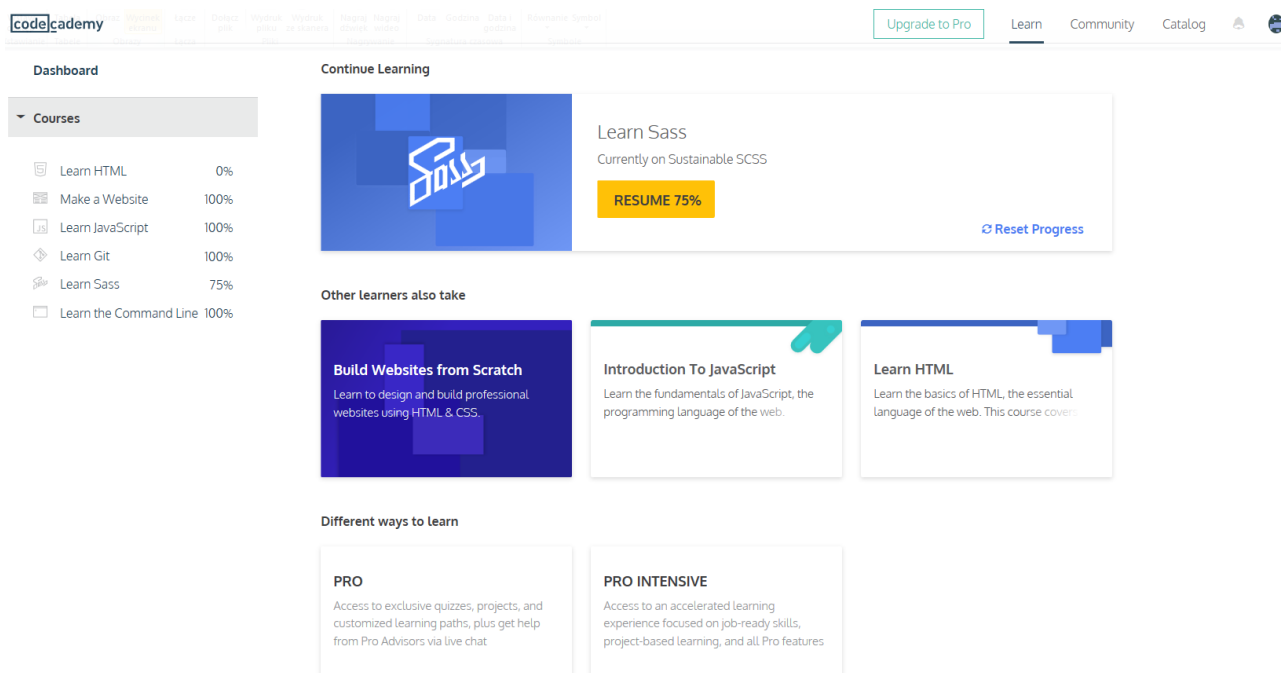


Рисунок 1.2 – Скріншот панелі керування курсами Codecademy

*Duolingo* [4] – сервіс онлайн-вивчення іноземних мов. По суті, цей аналог і став натхненням для описаної системи, адже саме вона зацікавила своїм алгоритмом підбору завдань з урахуванням індивідуальних навичок студентів. Для планування навчання створили та використовують метод регресії напівабування [5], який базується на поєднанні методів аналізу та обробки інформації із психологічною теорією кривою теорією кривої забування та технікою розподілених інтервальних повторень [6]. Ця модель дала поштовх до створення власної системи та пошуку моделей та методів планування тестування для підтримки запам'ятовуваності матеріалу, але яка б також мала функціонал системи дистанційного навчання, тобто, зробити певний симбіоз. На рисунку 1.3 нижче продемонстровано головну сторінку з тренуванням навички для користувача у сервісі Duolingo. Порівняно з попереднім аналогом, тут користувацький інтерфейс можна назвати авторським, оскільки багато дрібних елементів мають дуже специфічний, але одночасно й практичний дизайн. Щодо розміщення елементів, то візуально так само сторінка не виглядає перевантаженою, є основна секція, присутні бічні допоміжні секції та панель навігації зверху з основними елементами. Всі ці компоненти гармонійно розташовані на сторінці, а користувач отримає задоволення від роботи з даним сервісом.

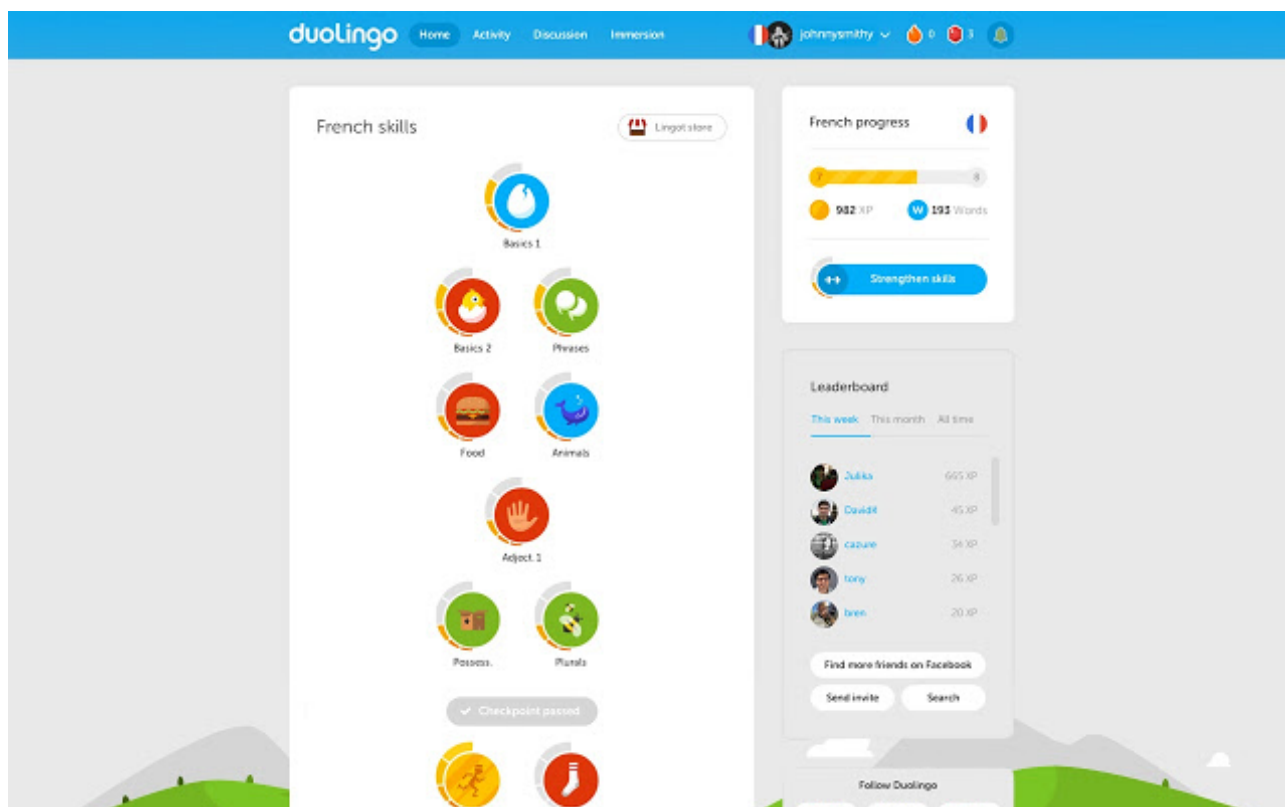


Рисунок 1.3 – Скріншот сторінки з навичками Duolingo

Також, окрім наведених вище прикладів, приділена увага системам дистанційного навчання. Сьогодні широко використовується велика номенклатура систем дистанційного навчання та управління дистанційним навчанням як з відкритим кодом (умовно безкоштовних), так і платних, широковживаних та вузькоорієнтованих [7].

*ATutor* – модульна система дистанційним керуванням навчанням з відкритим кодом. Поширюється на основі GNU General Public License. Для установки необхідно мати комп'ютер з веб-сервером Apache 1.3.x, PHP версії > 4.2.0 та MySQL версій > 3.23.x і > 4.0.12 (версії 4.1.x і 5.x офіційно не підтримуються). Система розроблена із врахуванням доступності та можливістю адаптації за бажанням користувача. Щодо операційної системи сервера, обмежень немає – система є кросплатформеною [8]. На рисунку 1.3 продемонстровано головну сторінку даної системи. Як видно, інтерфейс вже виглядає більш нагромадженим і трохи застарілим, але типовим як для системи дистанційного навчання.



Рисунок 1.4 – Скріншот головної сторінки ATutor

*Claroline* – платформа дистанційного навчання та електронної діяльності з відкритим кодом. Аналогічно з ATutor, поширюється на основі GNU General Public License. Сумісна з такими операційними системами, як Linux, Mac і Windows. Забезпечує інтуїтивно простий інтерфейс для адміністрування. В основу організації Claroline LMS покладено концепцію просторів, пов'язаних з курсом чи педагогічною діяльністю. Кожен з просторів забезпечений інструментарієм для створення, організації та управління навчальними матеріалами; можливостями для забезпечення взаємодії між користувачами тощо [9]. Як і у попереднього аналога, інтерфейс виглядає досить типово як для системи дистанційного навчання, але не таким нагромадженим, хоч і теж застарілим. Варто зазначити, що існує похідна версія Claroline Connect, яка є більш актуальною і має відповідний сучасний дизайн. Але вона не є поки що достатньо поширена серед навчальних закладів. Скріншот сторінки оголошення з системи наведено на рисунку 1.5 нижче.



Рисунок 1.5 – Скріншот сторінки оголошення Claroline

*Live@EDU* – система дистанційного навчання, виконана з використанням технології Active Server Pages на платформі Microsoft. Для установки та коректної роботи системи серверна частина повинна бути забезпеченою ОС Microsoft Windows NT Server 4.0, базою даних Microsoft SQL Server 7.0 та Microsoft Internet Information Server 4.0. Клієнтська частина повинна мати установлену ОС, яка забезпечує доступ до мережі Інтернет та браузер, що обслуговує протокол HTTP версії 3.0, а також програмне забезпечення для перегляду і створення лекційних матеріалів. Система інтегрується з іншими сервісами, найбільше з Microsoft Outlook. Також є доступ до різних сервісів – онлайн-версія PowerPoint, календар, хмарне сховище [7]. Але система вже є застарілою. Інтерфейс продемонстрований на рисунку 1.6 нижче. Як видно, все інтегровано в одному місці, Outlook, але при цьому нагромаджено і виглядає застарілим. Варто зазначити, що Microsoft має та розвиває інший актуальний сервіс Azure, який також дає змогу проходити як курси, так і використовувати цю платформу для власних цілей. Це все працює за хмарними технологіями та віртуальними машинами, які можна створити у будь-який момент. Більш докладно розглядатися даний сервіс не буде, оскільки курси у ньому не займають важливу

позицію, це також повноцінний робочий інструмент для розробників чи тестувальників.

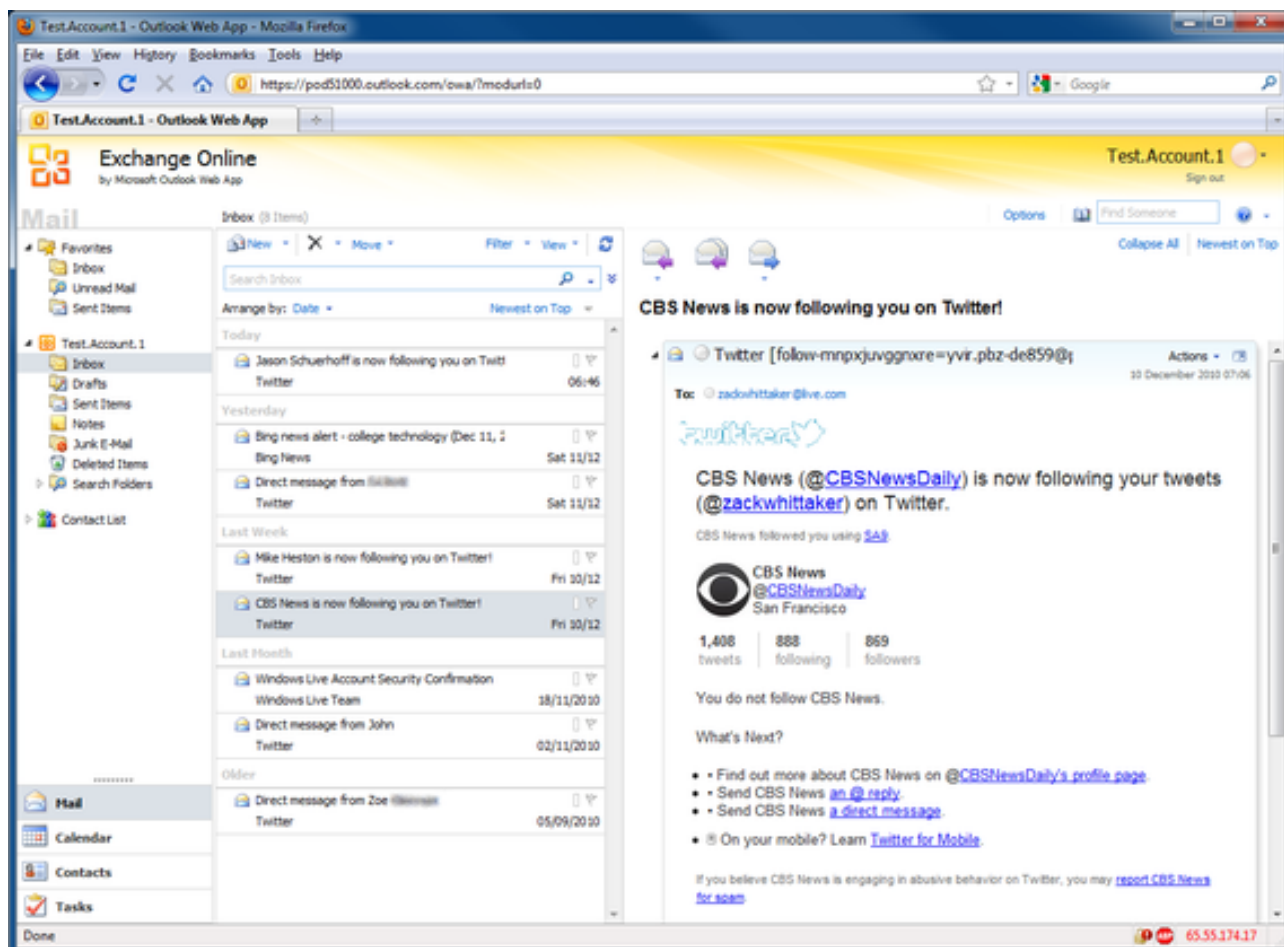


Рисунок 1.6 – Скріншот інтерфейсу Live@Edu в програмі Microsoft Outlook

*eFront* є новим поколінням систем електронного навчання, яка об'єднує в собі функції системи управління навчанням та системи створення та управління навчальних матеріалів. Використовується для організації навчального процесу у навчальних закладах, а також для підвищення кваліфікації, атестації та відбору працівників у різномасштабних організаціях. Система побудована на трьох типах користувачів – Адміністратор, Викладач та Студент [10]. На рисунку 1.7 нижче продемонстровано головну сторінку системи. Як видно, інтерфейс простий, а отже – зрозумілий. Також розміщення елементів інтерфейсу є типовим як і в інших систем дистанційного навчання. Система постійно розвивається, відбуваються оновлення, а отже, це означає, що вона є актуальною, незастарілою.



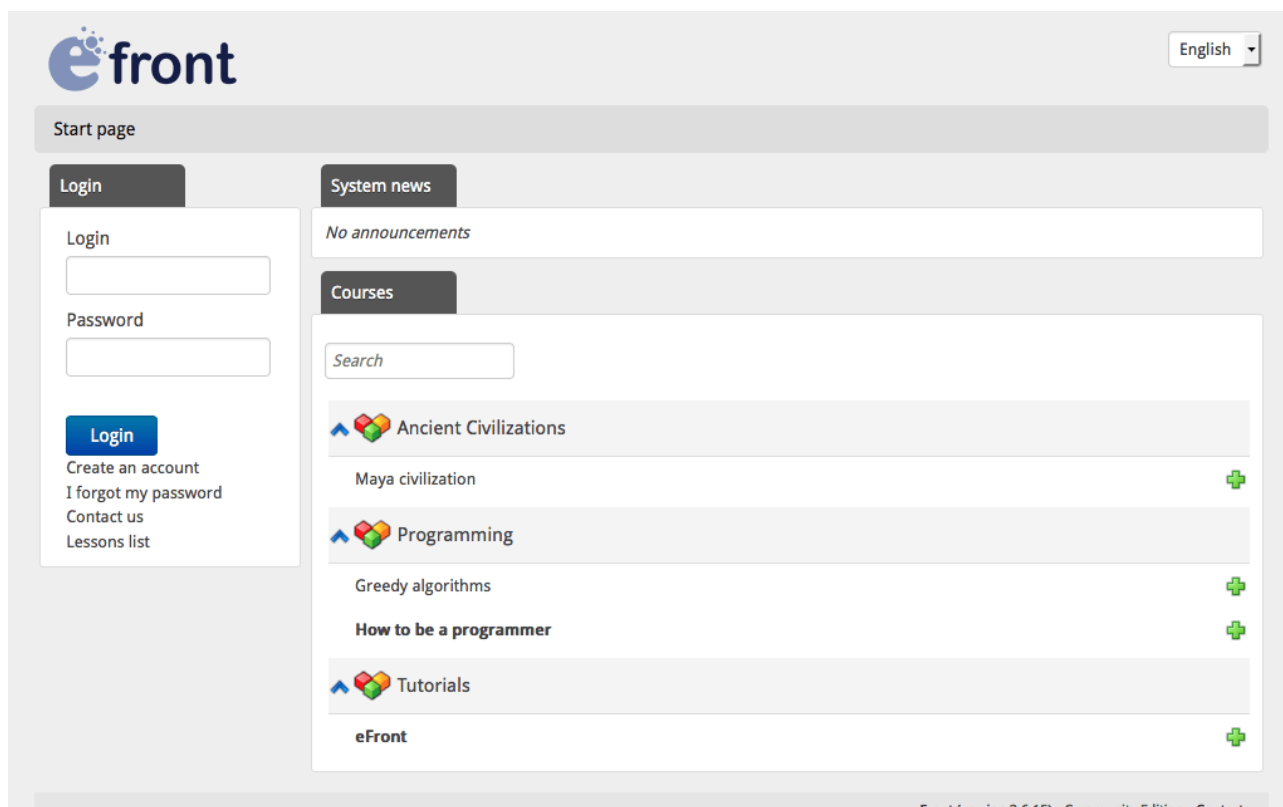


Рисунок 1.7 – Скріншот інтерфейсу системи eFront

*Moodle* (Modular Object-Oriented Dynamic Learning Environment) – пакет модульного програмного забезпечення з відкритим кодом (ліцензія GNU GPL), який призначений для створення курсів дистанційного навчання та web-сайтів. Ця програма управління дистанційним навчанням орієнтована на взаємодію між викладачем та студентом, також використовується для підтримки очних курсів. Moodle може бути встановленим на будь-який комп'ютер, який підтримує PHP та роботу із СУБД MySQL, PostgreSQL, Microsoft SQL Server; програмне забезпечення є кросплатформеним [11]. В основу проекту покладено п'ять принципів, які об'єднані спільною назвою “соціальний конструктивізм”:

- в сучасному навчальному середовищі ми всі одночасно є потенційними вчителями та студентами;
- ми успішні в навчанні, особливо тоді, коли намагаємось створити щось чи пояснювати щось людям;
- великий внесок в навчання роблять спостереження за роботою наших колег;
- розуміння інших дає змогу вивчити їх більш індивідуально;



- навчальне середовище повинне бути гнучким, забезпечувати учасникам навчального процесу простий інструмент для реалізації їхніх навчальних потреб [12].

На рисунку 1.8 продемонстровано інтерфейс системи дистанційного навчання Moodle. Як і попередній аналог, система досі підтримується і оновлюється. Вона є світовим лідером серед СДН і по кількості інсталяцій, і по кількості користувачів.

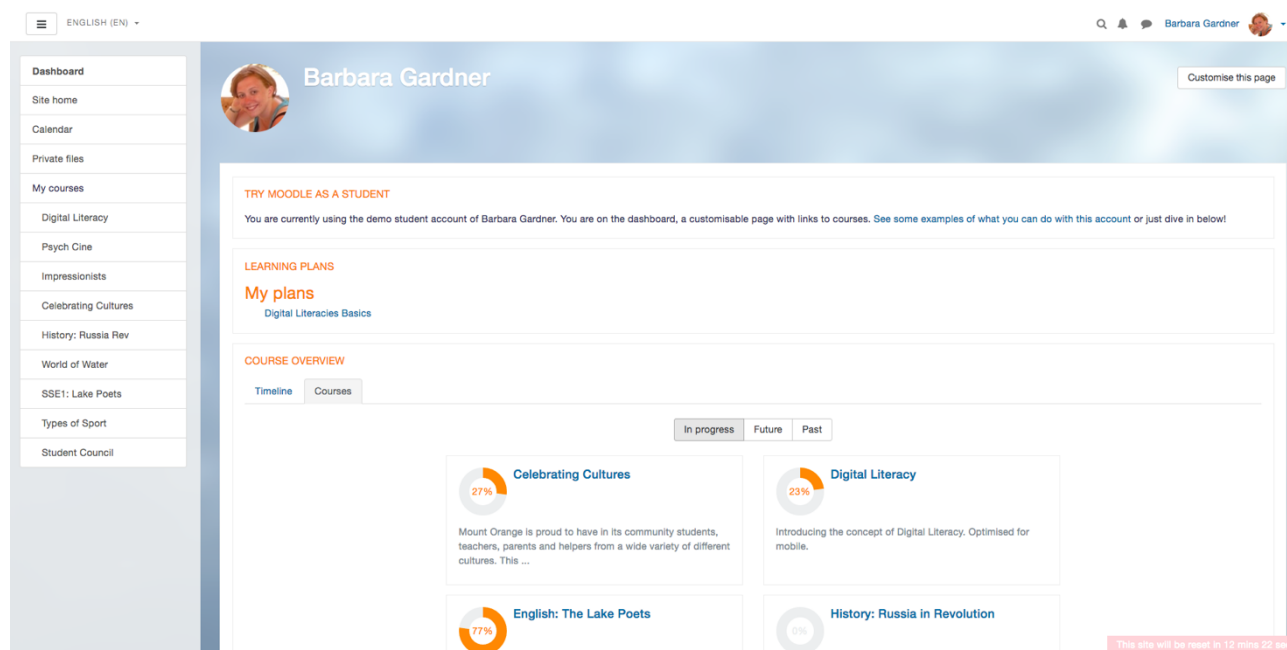


Рисунок 1.8 – Скріншот інтерфейсу системи Moodle

*SharePointLMS* – система дистанційного навчання, розроблена на потужній багатofункціональній платформі MS Office SharePoint Server 2007. Є комплексним рішенням, яке об'єднує всіх користувачів (викладачі, студенти, адміністратори тощо) у єдиний інформаційно-навчальний простір та забезпечує інструментарій для спільної роботи. На відміну від Moodle, Claroline та ATutor, система є платною. Використовується не лише навчальними закладами та центрами навчання, а й підприємствами, організаціями, державними структурами [13]. На рисунку 1.9 продемонстровано інтерфейс системи, який є також досить сучасним і простим. Він може трохи видозмінюватися залежно від постачальника-компанійона даної системи.

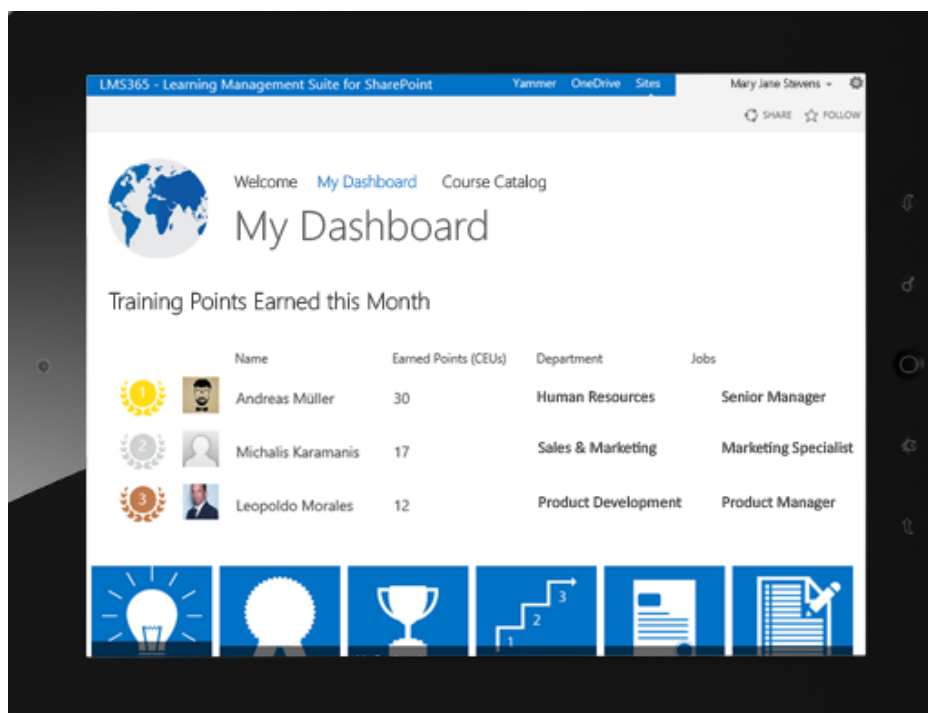


Рисунок 1.9 – Скріншот інтерфейсу системи SharePointLMS

Це перелік не всіх аналогів, але вище згадано найбільш поширені з них та використовувані в Україні та всьому світі. Найголовнішою відмінністю є не тільки відсутність підтримки різних локалізацій інтерфейсу, але й відсутність простого та зрозумілого інтерфейсу, а також підбору завдань для кожного користувача (не враховуючи наведений сервіс Duolingo, яка навпаки не є повноцінною системою дистанційного навчання за визначенням). Мета ж цієї системи – створити комфортні умови для підвищення запам'ятованості навчального матеріалу. Система постає у вигляді симбіозу системи дистанційного навчання та системи планування навчання з урахуванням індивідуальних особливостей студентів, заснована на методі інтервальних повторень. Щодо ключових характеристик, навчання на сервісі буде безкоштовним, у застосуванні будуть передбачені добровільні пожертвування. Питання монетизації (реклама, додаткові можливості користувача) – на даному етапі не розглядається. Отже, система буде безкоштовною і надаватиме користувачеві потрібні для запам'ятовування матеріалу завдання.

### 1.3 Рішення з інформаційного забезпечення

Для коректного функціонування системи необхідні вхідні дані у вигляді форматованої статистики користувачів, засоби для зберігання, виокремлення та аналізу цієї інформації, а також для індивідуального формування нових чи повторних

тестів для кожного окремого студента. Оскільки дані збираються від різних користувачів (клієнтів), для системи буде використана клієнт–серверна архітектура. Цільовою платформою обрано веб-застосування, оскільки воно буде підтримувати як версію для повноцінних комп’ютерів, так і для мобільних пристроїв. Через це було вирішено не витрачати додаткові ресурси на окремі мобільні застосування для різних мобільних операційних систем (iOS, Android). Це суттєво знизило витрати на розробку розглянутої системи. Також варто зазначити, що також можливо розглядати варіант інтеграції або реалізації даної системи у вигляді плагіна для вже використовуваної системи дистанційного навчання – Moodle [14].

Структура майбутньої бази даних, яка використовуватиметься для системи, що розробляється, наведена у вигляді ER-моделі у графічному матеріалі, на рисунку Додатка А.3.

#### **1.4 Цілі та завдання системи**

Враховуючи описані вище проектні рішення з розробки системи, можна визначити призначення системи, яка розглядається у даній магістерській дисертації. Система дистанційного адаптивного навчання призначена для організації процес підтримки методики навчального процесу, заснованої на методі інтервальних повторень. Це досягається шляхом планування навчання з урахуванням індивідуальних особливостей студентів на основі статистичної інформації.

Цілями даної системи є:

- полегшити процес запам’ятовування нового матеріалу;
- виявити матеріал, який сповільнює процес навчання і зробити відповідні висноки;
- створити індивідуальний навчальний план для студентів.

Для досягнення поставлених цілей необхідно вирішити наступні завдання:

- планування навчання та підбір завдань на основі успішності користувача (висновки про успішність робиться за допомогою обробки статистичної інформації);
- можливість перегляду статистичної інформації по всім користувачам і окремо по обраному користувачу для адміністратора та аналітика.

### **Висновок до розділу**

У даному розділі розглянуто та описано систему дистанційного навчання, що підтримує методику навчання, яка заснована на методі інтервальних повторень.

У ході опрацювання даного розділу описано бізнес-процеси для оглянутої системи, виконано опис процесу діяльності системи, розглянуто актори та функції, взаємозв'язок між користувачами та системою, проведено ознайомлення з самою структурою бізнес-процесів. На основі цього побудовано структурні схеми діяльності системи до оптимізації та після, модель варіантів використання.

Також велика частина роботи була пов'язана з оглядом існуючих аналогів. Опрацьовано багато літературних джерел та розглянуто більше десяти систем, які можна назвати аналогами до створюваної системи. Детально оглянуті переваги та недоліки цих систем – доступність, складність підтримки, простота інтерфейсу, наявність локалізації тощо.

У кінці розділу також було описано рішення з інформаційного забезпечення, в якому визначена архітектура майбутньої системи, якій присвячено окремий розділ 3. Також було створено ER-модель, яка показує структуру для бази даних, яка необхідна для проекту.

У результаті цим досягнуто те, що були визначені складові майбутньої системи, її бізнес-процеси та інші проектні рішення, отже, по суті, створено проектування системи, яке можна в подальшому використати для програмної реалізації.

## **2 МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ ПРОЦЕСІВ ТЕСТУВАННЯ В НАВЧАННІ**

### **2.1 Огляд наявного навчального процесу («як є»)**

Для того, щоб перейти до розробки зазначеного методу, необхідно спочатку оглянути навчальну методику [2]. У даній дипломній роботі описано приклад наявної реальної навчальної методики в одному з провідних навчальних закладів України. У Національному технічному університеті України «Київський політехнічний інститут імені Ігоря Сікорського» у Робочому навчальному плані зазначаються дисципліни за курсами (роками навчання), кількість годин/кредитів аудиторних занять і кількість годин для самостійного навчання. Також, там враховуються модульні (тематичні) контрольні роботи, екзамени, курсові, розрахункові роботи тощо. Семестр складається з 18 тижнів, здебільшого всередині нього є 2 модульні контрольні роботи. Ці роботи дозволяють перевірити засвоєння знань за модулем (темою), а також носять характер певної підготовки до екзамену. Ці модульні контрольні роботи не мають обмежень у своєму вигляді – це може бути розгорнута задача, набір завдань, есе тощо. Але зараз вже існують різні системи дистанційного навчання і одна з таких – Moodle – система дистанційного навчання, яка крім іншого допомагає проводити тестування та оцінювання знань студентів. Дана платформа стала досить популярною серед викладачів у нашому університеті. Враховуючи також час, який виділяється самостійному навчанню, викладачі також стали зараз на власний розсуд давати самостійні тести, які за об'ємом є набагато менші за модульну роботу, але в основному не мають ні обмежень по часу, ні у кількості спроб. Для студентів мотивацією є додаткові бали, які вони можуть заробити за проходження цих тестів. І це дійсно хороша практика, адже студенти отримують змогу частіше побачити певні питання за темою, пригадати пройдений матеріал і перевірити себе тестуванням. Викладач не витрачає час на проведення, перевірку та оцінювання тестів – усе це робить система. Найбільші організаційні витрати часу тільки перед першим роком застосування.

На прикладі конкретного предмету можна описати семестрову діяльність таким чином: вона складається з трьох вкладених циклів. Перший цикл – тижневий,

виглядає він так: викладач проводить лекцію, студент проходить самостійний тест, а Moodle оцінює цей тест і заносить у базу даних оцінки як за окремі питання, так і загальну оцінку за тест (3-4 питання). Наступний цикл – модульний. Він має декілька вкладених тижневих циклів, здебільшого це 6-8 тижнів, що складає 2 модульні тести за семестр. Виконання модульного тесту у середньому з 30 питань виглядає наступним чином: викладач налаштовує модульний тест, студент проходить модульний тест, а Moodle відповідно так само оцінює відповіді студентів. Третій і останній цикл – семестровий. На прикладі конкретної дисципліни він має два вкладені модульні цикли. Підсумкова оцінка за дисципліну формується так: викладач організовує екзамен, студенти виконують екзаменаційну роботу, а Moodle обчислює підсумкову оцінку, враховуючи семестрові бали та бали за екзамен.

Варто зазначити, що ми не розглядаємо практичні/лабораторні роботи та інші види діяльності за дисципліною. Основний акцент в описаній методиці навчання стоїть на самостійних тестах. Інтуїтивно зрозуміло, що їхня присутність іде на користь, адже це більш часте повторення матеріалу. Але цього разом не достатньо і ми вирішили знайти підхід/алгоритм, за яким можна було б визначати найкращі моменти проведення тестування.

## **2.2 Огляд задач оптимізації навчання і методів їх розв'язання іншими дослідниками**

### **2.2.1 Огляд досліджень технік запам'ятовування**

Для проведення досліджень потрібно було здійснити огляд вже існуючих методів, літератури, пов'язаної з тематикою випробувань. Все почалось з дослідження основ механізмів людської пам'яті.

З опрацьованих джерел, перші дослідження почалися з питань, а як все-таки ефективніше працювати над запам'ятовуванням матеріалу. Фундамент усім наступним дослідам заклав німецький вчений Герман Еббінгауз, який у ще у 1885 році відкрив таке явище як «крива забування». Крива забування або крива Еббінгауза була отримана внаслідок експериментального вивчення пам'яті німецьким психологом. Еббінгауз був захоплений ідеєю вивчення «чистої» пам'яті - запам'ятовування, на яке не впливають процеси мислення. Для цього ним було

запропоновано метод заучування безглузвих складів, що складаються з двох приголосних і гласної між ними, що не викликають жодних асоціацій за змістом (наприклад, бов, гіс, лоч тощо) [15].

В ході дослідів було встановлено, що після першого безпомилкового повторення серії таких складів забування йде спочатку дуже швидко. Вже протягом першої години забувається до 60% всієї отриманої інформації, через 10 годин після заучування в пам'яті залишається 35% від вивченого. Далі процес забування йде повільно, і через 6 днів в пам'яті залишається близько 20% від загального числа спочатку вивчених складів, стільки ж залишається в пам'яті і через місяць. Графічно крива забування представлена на рисунку 2.1.

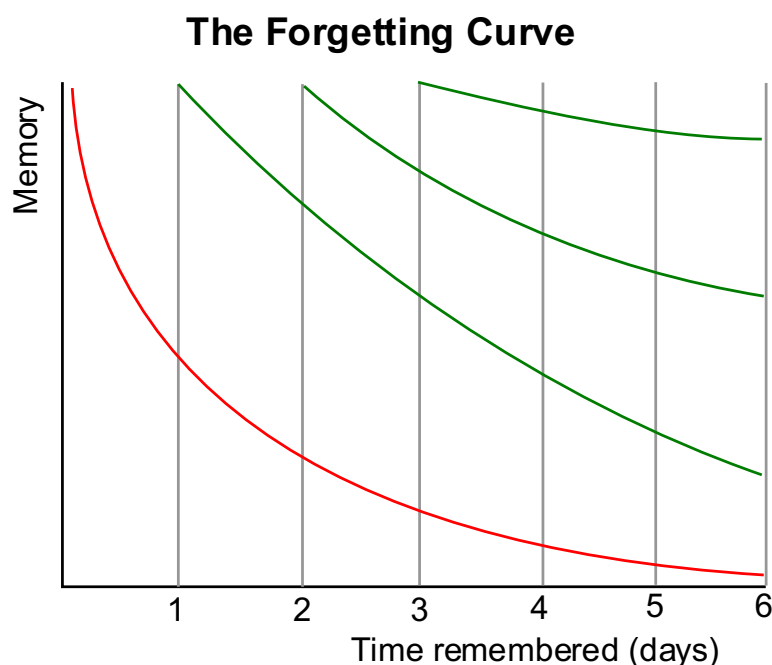


Рисунок 2.1 – Графічне представлення кривої забування Еббінгауза

Висновки, які можна зробити на підставі даної кривої, полягають в тому, що для ефективного запам'ятовування необхідне повторення завчено матеріалу.

Ідея, що розподілені повторення можна використовувати для поліпшення процесу навчання, вперше була запропонована в книзі «Психологія навчання» (англ. *Psychology of Study*), написаної професором Алеком Мейсом в 1932 році. У 1939-му Спітцер (Spitzer) протестував ефект методики на студентах в Айові. [16] Спітцер досліджував метод на більш ніж 3600 студентах і довів його ефективність. Але ця рання робота залишалася непоміченою до кінця 60-х, коли когнітивні психологи, особливо Ландуаер і Бйорк (Landuaer & Bjork) [17], а також Мелтон, [18]

досліджували маніпуляцію з інтервалом повторення як спосіб поліпшити запам'ятовування.

Приблизно в той же час Пол Пімслер став першопрохідцем в практичному застосуванні теорії інтервальних повторень для вивчення мов. Ця теорія вводила градуйовані розподілені повторення – це укорочений за часом варіант інтервальних повторень, опублікований Пімслером в 1967 [19]. Дана методика використовується в системі вивчення мови Пімслера, і особливо підходить для програмованих аудіопрограм через дуже короткі (вимірювані в секундах або хвилинах) інтервали між першими кількома повтореннями, на відміну від інших форм інтервальних повторень, які можуть не вимагати такої точності. Інтервали, опубліковані в його роботі, складали: 5 секунд, 25 секунд, 2 хвилини, 10 хвилин, 1 годину, 5 годин, 1 день, 5 днів, 25 днів, 4 місяці, 2 роки. Градуйовані розподілені повторення – складна назва для дуже простої теорії пам'яті. Жоден інший аспект вивчення іноземної мови не важливий настільки, як пам'ять, хоча до роботи доктора Пімслера ніхто ще не знаходив більш ефективного способу для створення мовної пам'яті. У своєму дослідженні доктор Пімслер описав, як довго студенти утримують в пам'яті нову інформацію та через які інтервали їм потрібно нагадувати про неї. Якщо інформація повторюється занадто рано або занадто пізно, вони забувають її. Це відкриття дозволило йому створити точний графік того, коли та як повинна подаватися інформація.

Перевіряючи програми, засновані на ідеях Післера, з секундоміром в руках, можна переконатися, що інтервали не завжди точно збігаються з вищевказаними, що пов'язано з існуванням верхніх і нижніх порогів. Схожий принцип (градуйовані інтервали з верхніми і нижніми порогами) використовується принаймні одним вільним програмним продуктом Gradint [20] для таймінгу аудіо-уроків.

У 1973 році Себастьян Лейтнер винайшов «Систему Лейтнера», універсальну систему навчання, засновану на картках. Вона більш адаптивна, ніж у Пімслера, оскільки інтервали можуть збільшуватися або зменшуватися в залежності від успішності студентів. Графічно ця система зображена на рисунку 2.2.



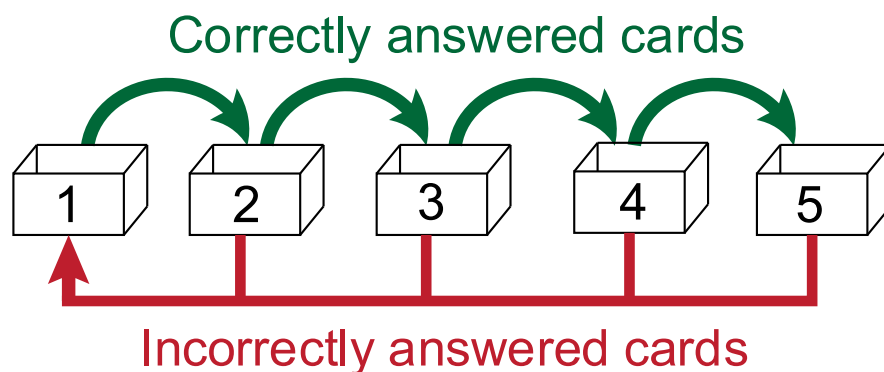


Рисунок 2.2 – Графічне представлення системи Лейтнера

У цьому методі так звані флеш-картки розсортовані в групи залежно від того, як добре учень засвоїв інформацію на кожній картці. Наприклад, при вивченні іноземної мови учень намагається згадати значення слова, написаного на флеш-картці. Якщо він згадує його, то картка перекладається в наступну групу. Якщо ж ні, то картка повертається до першої групи. Кожна наступна група повторюється через що збільшується інтервал. Даний метод може використовуватися як для вивчення слів іноземної мови, так і запам'ятовування іншої інформації.

Існує також трохи модифікований алгоритм, коли у разі неправильної відповіді картка повертається не в першу групу, а просто переходить у попередню. Графічно це зображено на рисунку 2.3 нижче.

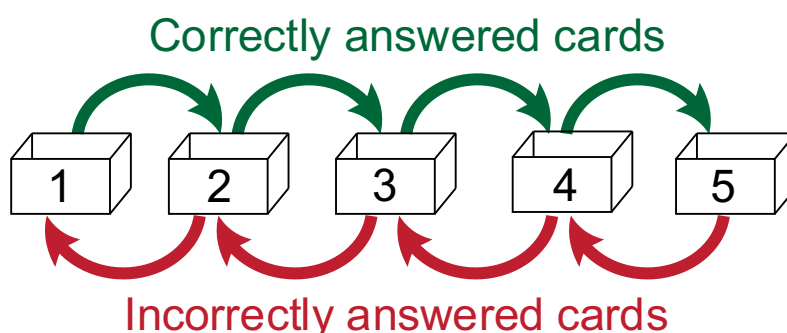


Рисунок 2.3 – Графічне представлення модифікованої системи Лейтнера

Найчастіше флеш-картки застосовуються при вивченні мови, але вони також підійдуть і для запам'ятовування будь-якої іншої інформації. Ви, наприклад, могли

бачити в американських фільмах, як готуючись виступити на весіллі або важливій зустрічі, автори записують на подібних картках тези свого виступу. Таким же чином готуються до іспитів, вчать програмування.

### 2.2.2 Застосування технік запам'ятовування у програмних продуктах

Система Лейтнера та метод інтревальних повторень покладені в основу багатьох програмних продуктів. Здебільшого всі вони стосуються вивчення іноземних мов. Нижче приведений перелік декількох таких програмних продуктів.

*Quizlet* [21]. Це програма, яку ви можете завантажити на свій смартфон (Android або iOS), а також користуватися нею в браузері. Багатьма вважається кращою програмою, що працює з флеш-картками. Володіє приємним дизайном, наявністю локалізації і зручною навігацією. Відстежує прогрес і відправляє нагадування. На рисунку 2.4 нижче зображено інтерфейс програми.

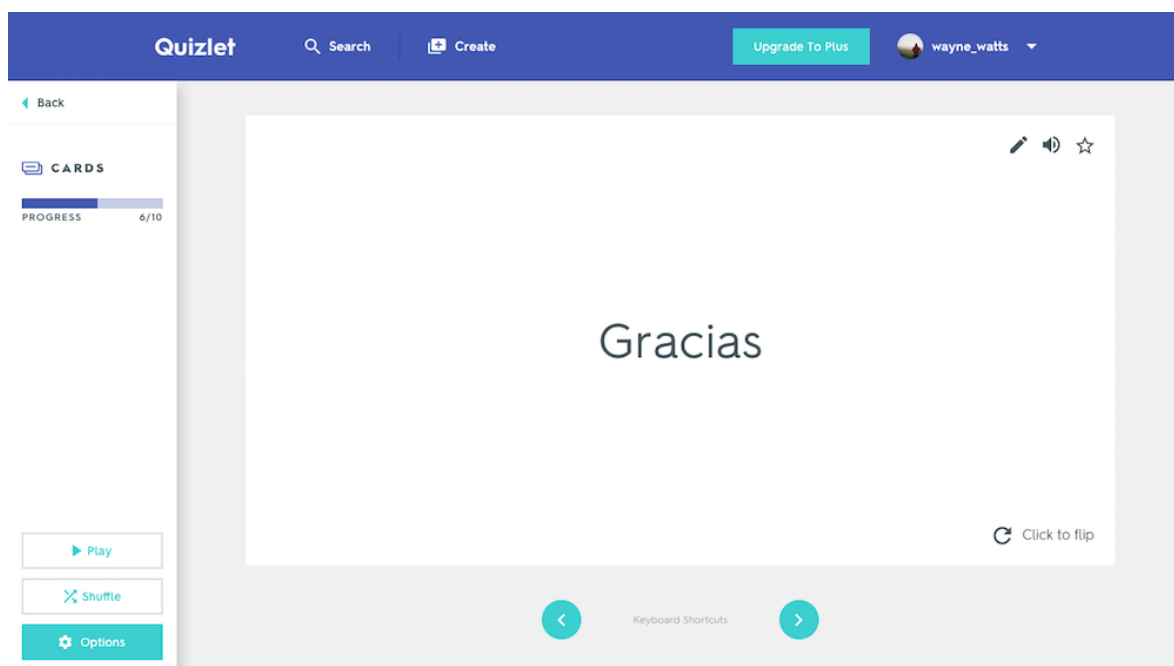


Рисунок 2.4 – Інтерфейс програми Quizlet

*StudyBlue* [22]. Дозволяє копіювати картки у інших людей. Ви можете знайти і створити групи, ділитися там відповідними картками з тими, хто вивчає той самий предмет, що і ви. Застосування дозволяє додавати аудіозаписи, картинки, а також влаштовувати вікторини. Доступно для Android, iOS і в браузері. Програма є платною. Інтерфейс зображено на рисунку 2.5 нижче.

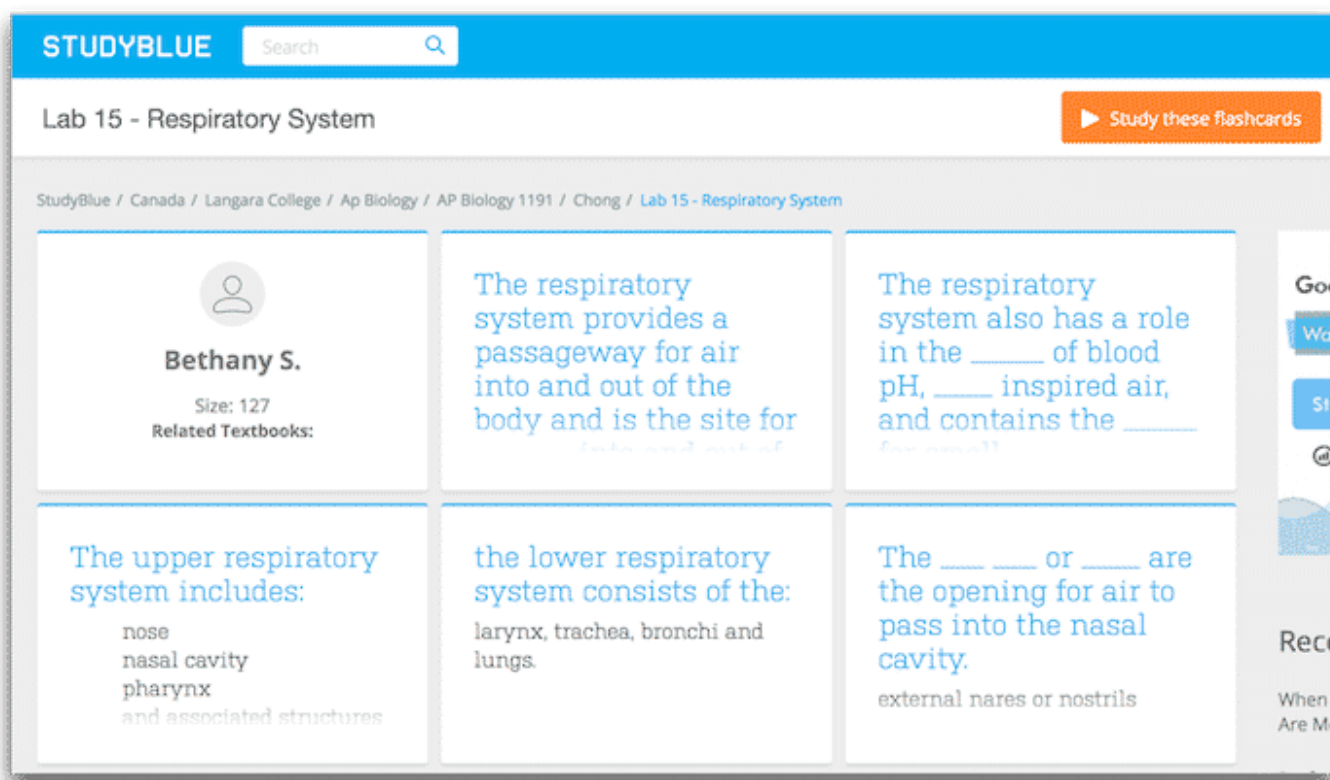


Рисунок 2.5 – Інтерфейс застосування StudyBlue

*StudyStack* [23]. Застосування, що використовує механізми гейміфікації [24]. Ви можете вчитися, розгадуючи кросворди і граючи в логічні ігри. Також можна знайти набори карток, влаштовувати вікторини і взаємодіяти з іншими людьми. Застосування доступне для Android, iOS, також є версія для браузера. На рисунку 2.6 продемонстрований інтерфейс даного застосування.

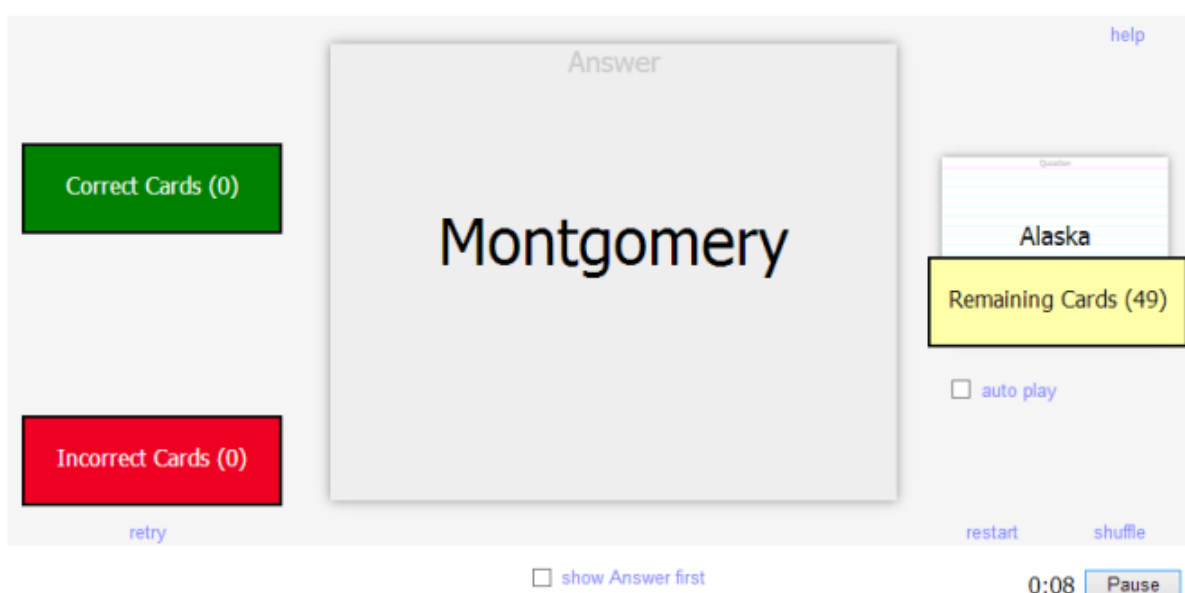


Рисунок 2.6 – Інтерфейс застосування StudyStack

*Brainscape* [25]. Застосування призначене тільки для карток: тут немає модних функцій, таких як вікторини або гри. Творці програми стверджують, що ви можете подвоїти швидкість навчання. Доступно для Android та iOS. Інтерфейс зображено на рисунку 2.7.

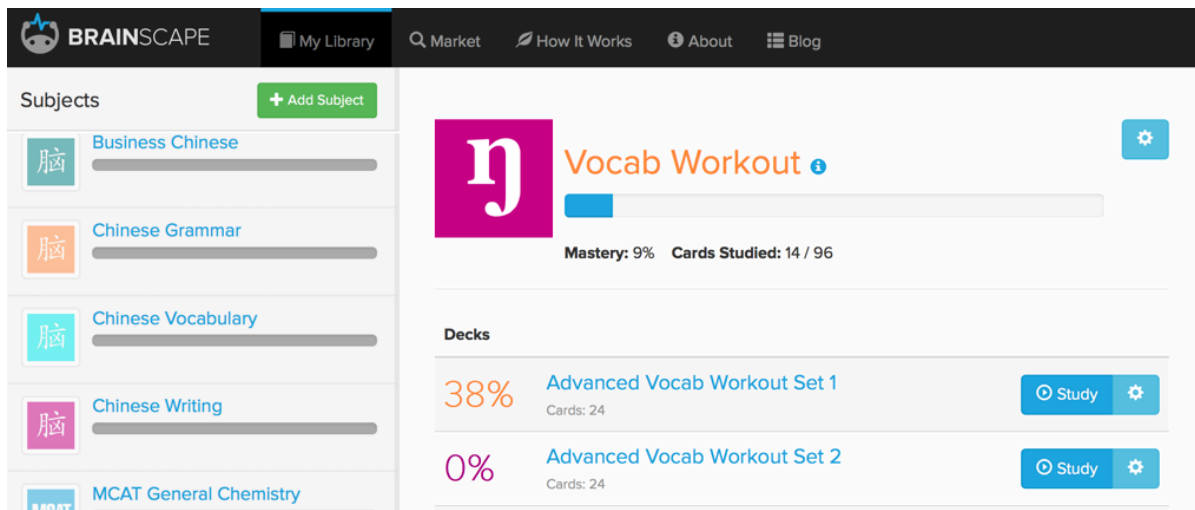


Рисунок 2.7 – Інтерфейс застосування BrainScape

*Anki* [26]. Колись це було найпопулярніше застосування в цьому жанрі, але з тих пір у нього з'явилося безліч конкурентів зі своїми перевагами. І більшість з них перевершує Anki по дизайну. І все ж застосування має серйозний функціонал і підтримує всі сучасні тенденції. Доступно для Android, iOS і браузера. Інтерфейс продемонстровано на рисунку 2.8.

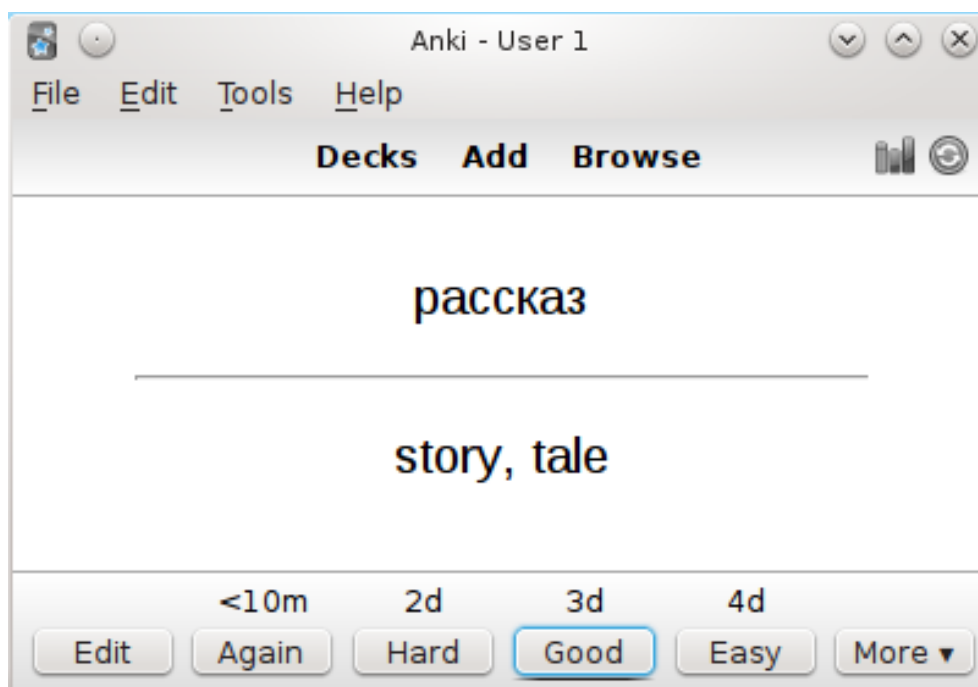


Рисунок 2.8 – Інтерфейс застосування BrainScape

### 2.2.3 Метод регресії періоду напівзабування

Метод регресії періоду напівзабування (Half-life regression method, HLR), запропонований фахівцями з Duolingo, є комбінацією психологічної теорії із сучасними методами машинного навчання [27].

Головною для теорії запам'ятовування є модель Еббінгауза, також відома як крива забування, яка була розглянута в розділі 2.3.1. Це означає, що з часом пам'ять експоненціально розпадається:

$$p = 2^{-\frac{\Delta}{h}}, \quad (2.1)$$

де  $p$  – імовірність правильного згадування елемента,

$\Delta$  – час, що пройшов з моменту останнього тренування,

$h$  – період «напівзабування», тобто час, за який людина забуде половину інформації після запам'ятовування, та ймовірність правильної відповіді дорівнює 0,5.

Графічно ця залежність зображена на рисунку 2.9 для випадку  $h = 1$ .

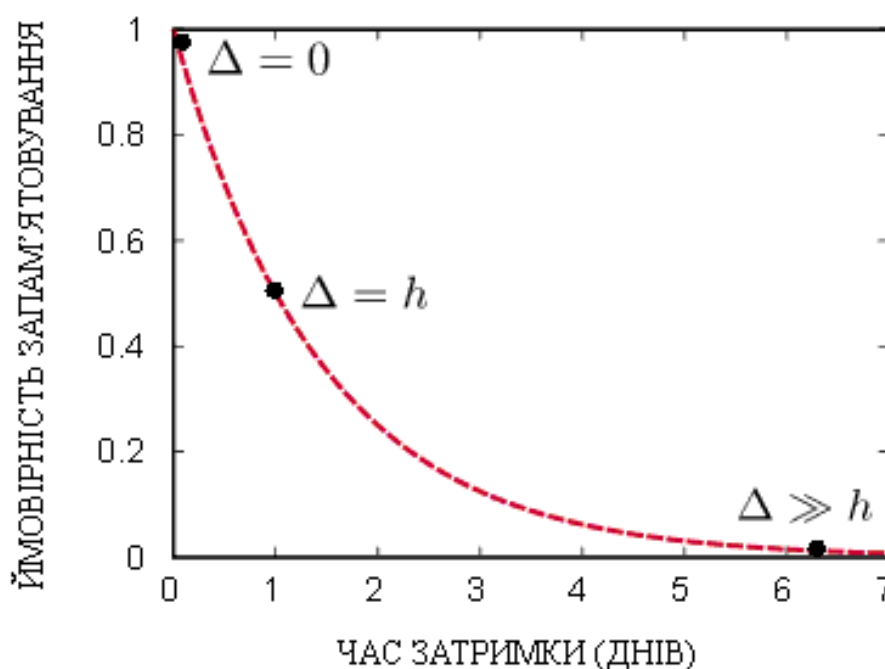


Рисунок 2.9 – Крива забування Еббінгауза

Тут можна виділити 3 випадки:

- $\Delta = 0$ . Це означає, що слово нещодавно практикувалося, тому  $p = 2^0 = 1.0$ , що відповідає думці про те, що воно є свіжим у пам'яті і його слід правильно згадати незалежно від періоду напівзабування;

- $\Delta = h$ . Час затримки між повтореннями дорівнює періоду напівзабування, а отже  $p = 2^{-1} = 0.5$ , і студент знаходиться на межі неможливості запам'ятатовування.
- $\Delta \gg h$ . Слово не повторювали давно відносно його періоду напівзабування, тому воно, ймовірно, було забуто, а отже,  $p \approx 0$ .

Формулу 2.1 можна представити в іншому вигляді. Нехай  $\mathbf{x}$  позначає вектор ознак, який підсумовує попереднє вивчення студентом певного слова, і нехай вектор параметрів  $\Theta$  містить ваги, що відповідають кожній змінній ознаки в  $\mathbf{x}$ . Згідно з припущенням, що період напівзабування повинен експоненціально збільшуватися з кожним повтором (звичайна практика в дослідженні відстані та затримки), будемо позначати ймовірний період напівзабування як  $\hat{h}_{\Theta}$ , який задамо як:

$$\hat{h}_{\Theta} = 2^{\Theta \cdot \mathbf{x}}. \quad (2.2)$$

Насправді методи Пімслера та Лейтнера можна інтерпретувати як особливі випадки для формули 2.2, використовуючи кілька фіксованих, підібраних вручну вагів.

Однак вони вирішили емпірично пристосуватися до вивчення даних про історію вивчення та вмістити доволіно великий набір цікавих особливостей. Припустимо, є набір даних  $D = \left\{ \langle p, \Delta, \mathbf{x} \rangle_i \right\}_{i=1}^D$ , що складається з сесій вивчення слова студентом. Кожен екземпляр даних складається зі спостережуваної швидкості виклику  $p$ , часу затримки  $\Delta$  з часу останнього повтору слова, та функціонального вектора  $\mathbf{x}$ , призначеного для персоналізації досвіду навчання. Мета – знайти найкращі ваги моделі  $\Theta^*$ , щоб мінімізувати деяку функцію втрат  $l$ :

$$\Theta^* = \arg \min_{\Theta} \sum_{i=1}^D l(\langle p, \Delta, \mathbf{x} \rangle_i; \Theta). \quad (2.3)$$

Для кращого розуміння представимо історію вивчення слова студентом за 30 днів на рисунку 2.10.

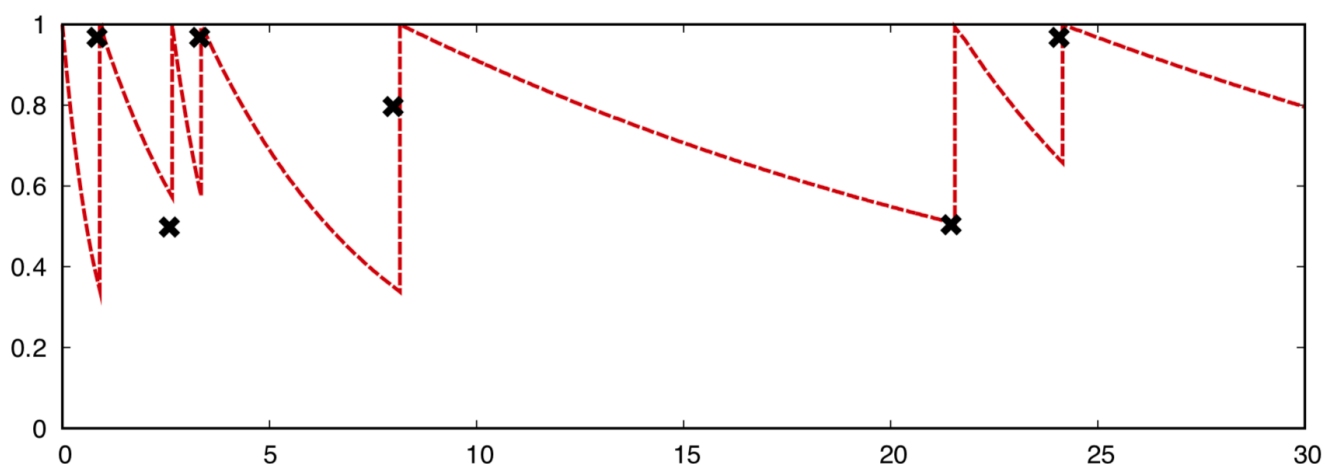


Рисунок 2.10 – Історія вивчення звуку, де вертикальна вісь – ймовірність згадування, горизонтальна – кількість днів

На рисунку кожен **x** вказує на екземпляр даних: вертикальна вісь – це очікувана ймовірність згадування  $p$  для кожного практичного заняття, а горизонтальна – час затримки  $\Delta$  між сеансами. Комбінуючи формули 2.1 і 2.2, передбачена моделлю ймовірність  $\hat{p}_\Theta = 2^{-\frac{\Delta}{\hat{h}_\Theta}}$  побудована у вигляді пунктирної лінії за часом (яка скидається до ймовірності 1,0 після кожного практикування, оскільки  $\Delta = 0$ ). Функція втрат під час навчання (формула 2.3) має на меті пристосувати прогнозовані криві забування до спостережуваних даних для мільйони статистичних даних щодо вивчення студентами вибраного слова.

Для цього обрали квадратичну функцію втрат, з  $L_2$ -регуляризацією, яка в своїй базовій формі задана наступним чином:

$$l(\mathbf{x}; \Theta) = (p - \hat{p}_\Theta)^2 + \lambda \|\Theta\|_2^2, \quad (2.4)$$

де  $\mathbf{x} = \langle p, \Delta, \mathbf{x} \rangle$  – коротке представлення для одиниці даних;

$\lambda$  – параметр для контролю регуляризації та попередження перенавчання.

Головна мета методу – мінімізації функції втрат з формули 2.4.

Щодо ефективності даного методу, то у Dualingo порахували його ефективність. На рисунку 2.11 зображено порівняння різних методів у вигляді діаграми.

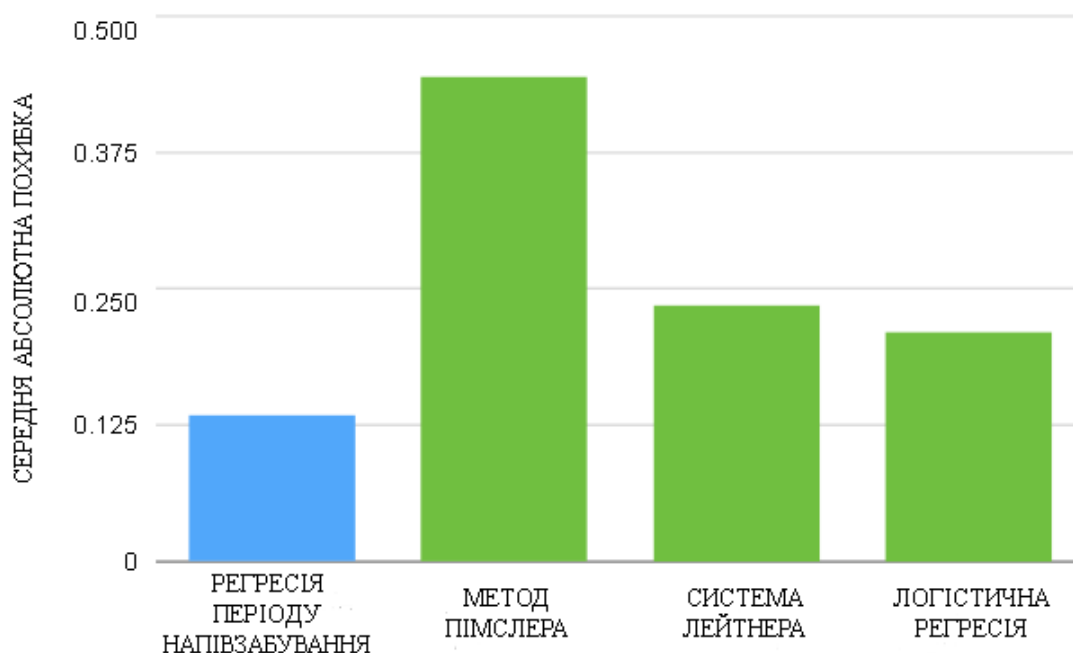


Рисунок 2.11 – Порівняння середньої похибки методів

Як видно з останньої ілюстрації, метод регресії напівзабування має найменшу середню абсолютну похибку.

З цього можна зробити висновок, що метод регресії періоду напівзабування продемонстрував високу ефективність, але він наразі працює тільки для вивчення іноземних мов і не всі розрахунки представлені розробниками. Отже, його не можна застосувати для описаного навчального процесу.

### 2.3 Удосконалення навчального процесу («як має бути»)

Проводиться удосконалення методики, розглянутої у розділі 2.1 даної роботи [2]. Спершу, потрібно визначити, скільки часу можна виділити на самостійні тести. Взявши дані того самого Робочого навчального плану, виходить, що для дисципліни із 120 годинами (4 кредитами) дається 66 годин самостійної роботи. Якщо виділити 10 відсотків на самостійне тестування, виходить, що на тиждень така дисципліна має 22 хвилини. Враховуючи розмір самостійного тесту (3-4 питання), то для об'єктивного оцінювання знань достатньо обмежити час на одне питання однією хвилиною, якщо воно не потребує розгорнутої відповіді. 18 хвилин з 22 можуть піти на вивчення та осмислення лекційного матеріалу. Отже, як видно з цих простих розрахунків, ми маємо достатньо часового ресурсу для проведення такого роду тестувань.



Дослідивши доцільність такого роду діяльності, ми дійшли до висновку, що самотійні тести дійсно покращують навчання чи запам'ятовування. Довелося трохи видозмінити вигляд описаного звичайного тижневого циклу. Алгоритм/методика полягає в тому, що якщо студент відповідає неправильно на якесь питання, то система запам'ятовує підкатегорію цього питання і до наступного тестування додасть ще одне питання з цієї підкатегорії. Якщо ж відповідь правильна, підкатегорія вилючається і не повторюється. Ми не збільшуємо інтервал, а саме вилучаємо через те, що це ресурсові обмеження – інакше об'єми тестів сильно б виростали, а з огляду на те, що викладається не одна дисципліна та в середньому щотижня проводиться лекція і пропонується новий самотійний тест – це необхідне обмеження. Отже, після оцінювання системою Moodle самотійного тесту, додається наступна умова: якщо на питання дана правильна відповідь, то питання з відповідної підкатегорії не додається до наступного тесту, в іншому випадку – після основних питань за наступною темою додаються питання-повторення із «забутих» підкатегорій. Таким чином, наступне тестування може не відрізнятися за розміром (всі відповіді правильні) або в гіршому випадку мати подвійний розмір (всі відповіді неправильні). Так, у найгіршому випадку існує ризик нагромадження, адже у разі неправильного ще одного повторення доведеться знову повторювати ці питання, а також включати наступні неправильні питання. Але, це буде значити, що такому студенту необхідно приділити більше уваги даній дисципліні або навчанню в цілому.

Що таке підкатегорії? Зараз питання у Moodle мають певні категорії, за якими і формується тест. Одна лекція включає дві категорії. Але доцільно поділити категорії на підкатегорії, щоб питання були максимально наближені один до одного. Звісно, здавалося б давати правильним на повторення ідентичне «провалене» питання, але множина питань дуже велика, тобто хоча і сам тест для одного студента має мало питань, але вони щоразу вибираються випадково з категорії, яка має багато питань; такий варіант має захист від випадку, коли студент запам'ятовує просто правильну відповідь, а не матеріал за темою. Тому повторне питання має бути максимально близьким до «проваленого», а саме, з тої ж під категорії.

## 2.4 Пошук оптимального терміну самостійного тестування та визначення періоду напівзабування

### 2.4.1 Задача пошуку оптимального терміну самостійного тестування

Протягом семестру, який триває 18 тижнів, після кожної лекції упродовж двох днів після неї в системі дистанційного навчання Moodle студенти проходять так званий «самостійний тест». Студенту дається одна спроба з трьох питань по матеріалу лекції. Після кожного питання система повідомляє студенту, чи є відповідь на питання правильною.

Ми намагаємось забезпечити максимальне запам'ятовування матеріалу на кінець певного фіксованого періоду, як-от місяць. В який момент протягом цього періоду доцільно застосувати самостійний тест?

Згідно кривій забування Еббінгауза, а також методу регресії напівзабування, трохи перетворимо формулу 2.1, і представимо в такому вигляді для проведення подальших розрахунків:

$$p(x) = 2^{-\frac{x}{h}}, \quad (2.5)$$

де  $p(x)$  – імовірність правильної відповіді;

$x$  – час, що пройшов після першого запам'ятовування;

$h$  – період «напівзабування», тобто час, за який людина забуде половину інформації після запам'ятовування, та ймовірність правильної відповіді дорівнює 0,5.

Також приймаємо, що після кожної вірної відповіді період напівзабування збільшується в  $\alpha$  разів [28]:

$$h_n = \alpha h_{n-1}, \quad n = 0, \dots, N, \quad (2.6)$$

де  $n$  – номер спроби (номер тесту);

$h_n$  – період напівзабування для спроби  $n$ ;

$h_0$  – початковий (мінімальний) період напівзабування.

На відміну від експериментів Еббінгауза, учасники яких запам'ятовували ланцюжки абстрактних символів, у навчальному процесі ми маємо справу з добре структурованим, осмисленим матеріалом, який добудовується до вже існуючої

системи знань у студента. Логічно передбачити, що період напівзабування збільшується не обов'язково у два рази, а з іншою невідомою нам швидкістю.

Нехай період тестування починається з нуля і закінчується в момент  $y$ . Тестування відбувається в момент  $x$ ,  $0 \leq x \leq y$ . Мета – максимум функції  $p(y)$  – імовірності правильної відповіді в кінці періоду. З урахуванням тестування в момент  $x$  є дві можливості:

- з імовірністю  $1 - p(x)$  студент відповів неправильно, процес забування продовжився, і в кінці періоду імовірність правильної відповіді становить

$$p^-(y) = 2^{-\frac{y}{h_0}};$$

- з ймовірністю  $p(x)$  студент відповів вірно, процес забування почався спочатку, період напівзабування збільшився в  $\alpha$  разів, в кінці періоду

$$\text{імовірність правильної відповіді } p^+(x) = 2^{-\frac{y-x}{\alpha h_0}}.$$

Для першого випадку крива забування, де демонструється залежність між ймовірністю згадування та кількістю часу після отримання матеріалу, наведена у у графічному матеріалі, на рисунку Додатка А.4.

Для другого випадку (вдале проходження) ця залежність виглядає інакше та наведена у графічному матеріалі, на рисунку Додатка А.5.

Критерій задачі пошуку оптимального терміну тестування має вигляд:

$$\max_x f(x) = \max_x ((1 - p(x))p^-(y) + p(x)p^+(x)). \quad (2.7)$$

Розкривши  $p(x), p^-(x), p^+(x)$ , отримаємо:

$$f(x) = 2^{-\frac{y}{h_0}} \cdot (1 - 2^{-\frac{x}{h_0}}) + 2^{-\frac{x}{h_0}} \cdot 2^{-\frac{y-x}{\alpha h_0}} = 2^{-\frac{y}{h_0}} \cdot (1 - 2^{-\frac{x}{h_0}}) + 2^{-\frac{(\alpha-1)x+y}{\alpha h_0}}. \quad (2.8)$$

В останньому виразі в (2.8) перша складова виражає процес забування при невдалому тесті, а друга складова – процес забування при вдалому тесті, причому вплив другої складової на результат переважає вплив першої складової, в чому можна переконатись, побудувавши графік функції, який наведено у графічному матеріалі, на рисунку Додатка А.6. Сумарний графік наведено на рисунку Додатка А.7.

Пошук оптимального  $x$  шляхом вирішення рівняння  $f'(x_{opt}) = 0$  дає загальний результат у формі:

$$f'(x) = \frac{\ln 2}{h_0} 2^{-\frac{x+y}{h_0}} - \ln 2 \frac{\alpha - 1}{\alpha h_0} \cdot 2^{-\frac{(\alpha-1)x+y}{\alpha h_0}}, \quad (2.9)$$

$$\frac{x_{opt}}{y} = \frac{a - \alpha}{\alpha - a(\alpha - 1)}, \text{ де } a = \log_2 \frac{\alpha - 1}{\alpha}. \quad (2.10)$$

Для  $\alpha = 2$  в результаті підрахунків отримано  $\frac{x_{opt}}{y} = -1$ , а для  $\alpha = 1,14$  результат

наступний:  $\frac{x_{opt}}{y} = -2,666$ .

Від'ємне значення  $x_{opt}$  означає, що оптимум не знаходиться в допустимій області  $0 \leq x \leq y$ . З цього випливає, що з урахуванням усіх факторів тест варто проводити чим раніше, тим краще, бажано відразу після лекції.

#### 2.4.2 Визначення періоду напівзабування

Окрім вирішення задачі оптимального терміну тестування можливо також розрахувати період напівзабування матеріалу в пам'яті. Для цього потрібно перетворити формулу 2.1, а саме логарифмувати її:

$$\log_2 p(x) = -\frac{x}{h}, \quad (2.11)$$

де  $p$  – імовірність правильної відповіді;

$x$  – час, що пройшов після першого запам'ятовування;

$h$  – період «напівзабування», тобто час, за який людина забуде половину інформації після запам'ятовування, та ймовірність правильної відповіді дорівнює 0,5.

Тепер з формули 2.11 можна виразити  $h$  через натуральний логарифм:

$$h = -\frac{x}{\log_2(p(x))}, \quad (2.12)$$

$$h = -\frac{x \cdot \ln(2)}{\ln(p(x))}. \quad (2.13)$$

Отже, завдяки розрахункам, за формулою 2.13 тепер можна розрахувати період напівзабування. Що і буде зроблено і описано у розділі 2.6 даної роботи.

## 2.5 Дослідження числових параметрів процесу запам'ятовування

В експерименті взяли участь близько 90 студентів, які проходили тест двічі, в період 1-2 днів після лекції і той самий тест за 7 днів після першої спроби. Були відкинуті одиничні спроби (ті, де максимум був отриманий з першої спроби і через це студенти не проходили повторно) та спроби, які, ймовірно, були некоректно збережені через проблеми з мережевим з'єднанням, і мали оцінки 1,25-2 бали при максимальній оцінці 4. В результаті кінцева для кінцевої вибірки були взяті дані 18 студентів.

Докладні результати для кожного студента можна переглянути у таблиці 2.1 нижче.

Таблиця 2.1 – Результати двох спроб одного тесту

Спроба	Оцінка за першу спробу	Оцінка за другу спробу
Користувач 1	3,00	4,00
Користувач 2	3,00	4,00
Користувач 3	4,00	4,00
Користувач 4	3,00	4,00
Користувач 5	3,00	4,00
Користувач 6	3,00	3,25
Користувач 7	4,00	4,00
Користувач 8	2,67	3,00
Користувач 9	3,00	4,00
Користувач 10	3,00	3,00
Користувач 11	3,00	4,00
Користувач 12	3,25	3,00
Користувач 13	4,00	4,00
Користувач 14	3,50	4,00

Спроба	Оцінка за першу спробу	Оцінка за другу спробу
Користувач 15	3,30	3,00
Користувач 16	4,00	4,00
Користувач 17	3,00	3,50
Користувач 18	3,00	4,00
<i>Середня оцінка</i>	<i>3,26</i>	<i>3,71</i>

Графічно результати експерименту наведені у вигляді діаграми у графічному матеріалі, на рисунку Додатка А.8.

Проаналізувавши отримані результати, дійшли до наступного: інтервал між повтореннями збільшується не в 2 рази, а повільніше, що пов'язано з тим, що нові знання добудовуються до існуючих, на відміну від експериментів Еббінгауза, де запам'ятовувались абстрактні ланцюжки символів.

Для цього випадку можна вже розрахувати значення  $\alpha$ . З таблиці вище потрібно просто порахувати співвідношення другої середньої оцінки до першої. Для даного експерименту співвідношення має вигляд:  $\frac{3,71}{3,26}$ . Внаслідок цих дій отримано

наступний результат:  $\alpha = 1,14$ .

Використовуючи формулу 2.13, проведемо розрахунок періоду напівзабування після двох послідовних самостійних тестувань зі збільшеними інтервалами. Отже, можна виділити два випадки для двох часових інтервалів.

- після першого тестування  $x$  дорівнює 1,5, оскільки після першого ознайомлення з матеріалом до першого самостійного тестування пройшло 1-2 дні (у середньому 1,5), а отже в такому разі період напівзабування дорівнює:

$$h = -\frac{1,5 \cdot \ln(2)}{\ln\left(\frac{3,26}{4}\right)} = -\frac{1,5 \cdot \ln(2)}{\ln(3,26) - \ln(4)} = -\frac{1,04}{1,18 - 1,39} = 4,95;$$

– після другого тестування  $x$  дорівнює 7, оскільки між першим і другим тестуванням у середньому пройшло 7 днів, і тоді період напівзабування

$$\text{дорівнює: } h = -\frac{7 \cdot \ln(2)}{\ln\left(\frac{3,71}{4}\right)} = -\frac{7 \cdot \ln(2)}{\ln(3,71) - \ln(4)} = -\frac{4,85}{1,31 - 1,39} = 60,625.$$

Отже, з отриманих результатів маємо, що період напівзабування після першого тестування становив майже 5 днів, а після другого – вже 60 днів.

З цього можна зробити висновок, що якщо з часом збільшувати інтервали між повтореннями, то більше інформації буде утримуватися в пам'яті людини. І хоча за попередніми розрахунками значення  $\alpha$  є меншим, то значення  $h$  навпаки сильно виростає зі збільшенням інтервалу.

Це призводить до тієї ситуації, яка стосується кривої Еббінгауза: найкраще знайти час на повторення посередині часової лінії, щоб робити менше спроб і збільшувати на більші проміжки часу інтервали між повтореннями. Але, при цьому, не варто брати максимальні інтервали, оскільки в такому випадку, шанс, що якась інформація залишиться в пам'яті дуже малий.

### **Висновок до розділу**

Даний розділ був присвячений методам і моделям пошуку оптимального терміну самостійного тестування. Вирішення даної задачі є необхідним для реалізації задуманої системи підтримки методики навчального процесу.

Для складання задачі розглянута наявна навчальна методика, реальний практичний приклад з навчального закладу. Для цієї методики також запропонована оптимізація – введення методу інтервальних повторень до самостійного тестування.

Також у процесі роботи були розглянуті вже існуючі методи, які могли б вирішувати цю задачу, джерела щодо історичного підґрунтя та базових основ для побудови постановки задачі. Фактично, основою задачі стала психолонічна теорія, яка була виведена у кінці 19 століття німецьким вченим Еббінгаузом. Окрім того, були розглянуті сервіси, які вже використовують метод інтервальних повторень або якісь суміжні з ним.

У результаті, завдяки виконаній роботі досягнуто те, що можна реалізувати програмно алгоритм, який вирішуватиме задачу і формуватиме індивідуальний план навчання для студентів.



## **3 ОПИС ПРОГРАМНОГО ТА ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ**

### **3.1 Засоби розробки**

#### **3.1.1 Вибір платформи для системи**

Враховуючи вищенаписане, зрозуміло, що система буде представлена у вигляді клієнт-серверного застосування. В даному випадку існує дві найбільш популярні опції: реалізація веб-застосування та реалізація мобільного застосування.

Тема вибору між мобільними і веб-застосуваннями була, є і буде актуальною. Популярністю користуються обидва варіанти, кожен з яких має як свої переваги, так і недоліки. В цілому все не складно: щоб точно визначитися з вибором між розробкою мобільних і веб-застосувань, потрібно розглянути кілька критеріїв.

В першу чергу це цілі, яких розробник хоче досягти (і функції, які повинно мати застосування), цільова аудиторія і ваш бюджет. Перш ніж безпосередньо перейти до розгляду мобільних і веб-застосувань, для початку давайте наведемо визначення веб-застосування.

Веб-застосування – це клієнт-серверне застосування, (клієнтом є браузер, а в якості сервера виступає веб-сервер), при якому зберігання даних здійснюється головним чином на сервері, а обмін даними відбувається через мережу. З цього випливає, що для роботи з веб-застосуванням користувачеві необхідний доступ до мережі Інтернет. Відмінною особливістю веб-застосування є масштаб: одночасно ним може користуватися велика кількість людей [29].

Однією з переваг вибору розробки веб-застосування є той факт, що користувачі не залежать від операційної системи, тому ці застосування кросплатформні.

Недоліком клієнт-серверного рішення є його залежність від мережі. Те саме стосується і мобільного застосування, якщо у ньому немає функції викачування даних. Для системи, що буде розглядатися, потрібне постійне оновлення даних – а отже, з'єднання з мережею, тому немає сенсу викачувати велику кількість інформації на пристрій.

Що стосується оплати для кінцевого користувача, то за роботу з веб-застосуванням інколи потрібно вносити щомісячну плату (найчастіше це незначна сума). Також одним з варіантів є здійснення оплати на основі проведених операцій.

Якщо користувач запізнюється з платежем, то доступ до роботи з веб-застосуванням закривається. Таким чином, клієнт як би прив'язується до постачальника: він не може відмовитися від його послуг (або не платити) і при цьому продовжувати користуватися застосуванням. Іноді веб-застосування надається безкоштовно, але користувачеві пропонуються преміум-функції, за доступ до яких необхідно платити.

Одна з безперечних переваг веб-застосувань полягає в тому, що вони не вимагають встановлення на смартфоні. Тобто пам'ять на пристрої вони не займають (тільки трохи кеша), на відміну від мобільних застосувань. Проте, необхідно відзначити, що в плані функціоналу, розробка мобільного застосування дає більше, ніж клієнт-серверна розробка, але зате витрат і часу на неї теж потрібно більше.

В цьому плані веб-застосування є хорошим вибором, особливо якщо вам потрібно щось дуже просте, що не вимагає широкого складного функціоналу, наприклад, новинна стрічка. Не слід забувати і той факт, що мобільне застосування практично завжди йде на додаток до веб-застосування, тобто компанія починає з розробки свого веб-застосування, і тільки потім переходить до розробки мобільного [30].

Також перевагою розробки веб-застосувань є те, що оновлювати їх можна завжди, в будь-який час, і це не викликає ніяких складнощів: після цього нова версія застосування буде доступна всім користувачам. З мобільними застосуваннями інша ситуація: при оновленні (наприклад, дизайну), внесення будь-яких поліпшень і розробці додаткових функцій виходить нова версія, яку користувачеві знову потрібно завантажувати з магазину. Проте, просування веб-застосування часто вимагає більших витрат, ніж для мобільного, хоча сама веб-розробка практично завжди виходить дешевшою (якщо брати до уваги схожий функціонал).

Тут вся суть в тому, що хорошою рекламою є самі магазини мобільних застосувань, в яких користувач, охочий встановити застосування, розглядає різні варіанти – порівнює відгуки, рейтинг продукту, а також бачить кількість завантажень. Звичайно, тут є і мінус в тому плані, що якщо застосування, скажімо так, не дуже хороше, то відгуки теж будуть відповідними.

Маркетинг є одним з найважливіших з етапів: перед розробкою мобільного застосування слід провести аналіз конкурентів, ринку, потреб вашої цільової аудиторії. Ідея повинна бути релевантною, а продукт – відповідати всім запитам користувачів, бути зручним у використанні тощо. Тоді ви отримаєте і позитивні відгуки, і скачування, і рейтинг: самі магазини мобільних застосувань стануть для вас найкращою рекламою. І в кінцевому підсумку на вас чекає успіх.

У той же час при просуванні веб-застосування часто необхідні значні витрати на пошукову оптимізацію (SEO) [31], і часто зекономлені на розробку кошти йдуть на маркетинг. Головним фактором є аудиторія: в магазини мобільних застосувань люди майже завжди заходять конкретно за тим, щоб завантажити застосування з потрібними функціями, і тут питання вибору: що їм сподобається, де відгуки краще, рейтинг вище і т.д. А для успіху веб-застосування часто необхідно вкладати кошти в його просування (з мобільними важко, але не до такого ступіню, там більше питання в правильному підході до розробки програми).

Ще однією перевагою розробки мобільних застосувань є те, що ви можете бути впевнені, що не скачаєте собі який-небудь вірус, тому що магазини застосувань ретельно все перевіряють перед тим, як зробити додаток доступним для скачування. Також мобільні застосування мають доступ до сховища пристрою (і багато чого іншого), на відміну від веб-застосувань, які запитують для цього певні права під час процедури встановлення. Ще однією великою перевагою розробки мобільного застосування є те, що ви можете "залізти в голову" користувача і міцно там влаштуватися. Що мається на увазі: мобільні застосування дозволяють постійно "бути на зв'язку" з вашими клієнтами. Ви, а не вони, встановлюєте контакт і підтримуєте його. Ви можете робити корисні розсилки, повідомляти про акції та знижки, вітати зі святами тощо – все, щоб збільшити ваші продажі і в той же час надати вашим користувачам відмінні послуги.

Також недоліками мобільних застосувань є те, що потрібно знову думати про цільову аудиторію. За своєю архітектурою вони можуть бути кросплатформними або ж написані тільки під відповідну операційну систему. Наразі найбільш поширені

мобільні операційні системи – Android та iOS. До того ж, мобільні застосування потребують більш глибоких знань та більше часу на реалізацію проекту.

Враховуючи обмеження у часі, та наявні навички, а також специфіку систему, то логічно все-таки зупинитись на реалізації веб-застосування. Також, варто зазначити, що сучасні веб-застосування чудово підтримують і мобільні пристрої, тобто вони є адаптивними до розмірів екрану. А отже, вибір такого застосування дозволить знизити витрати на розробку, оскільки воно покриє всі необхідні для системи пристрої.

### **3.1.2 Вибір мови програмування**

Отже, визначившись з цільовою платформою, необхідно вибрати мову програмування. Основний акцент робиться на клієнтську частину, а вже похідною стане серверна реалізація. Для такої організації оптимальним варіантом буде JavaScript. Хоча це і не найпростіша мова програмування, але вона має достатній потенціал для побудови такої системи. JavaScript забезпечує динамічність сторінки. Анімація, інтерактивні кнопки і графічні елементи: ми вже насилу уявляємо собі додатки без всього цього. З одного боку, JavaScript – це клієнтська мова. Клієнтом в цьому випадку є браузер, які інтерпретує код програми та виконує його. Інтерпретатор коду вбудований в усі популярні браузери, тобто в плані підтримки ніяких обмежень не існує. Але цю мову також можна використовувати і на серверній частині завдяки додатковим платформам, наприклад Node.js чи Firebase [32].

Наступним кроком необхідно вибрати фреймворк, який буде використовуватися для розробки веб-застосування.

### **3.1.3 Вибір фреймворка для розробки веб-застосування**

Найбільш популярними фреймворками для створення веб-застосувань на даний момент є React, Angular, Vue.js [33]. Між ними є відмінності, доволі суттєві. Проте, вони дозволяють вирішити ту саму задачу. Але проблема вибору при цьому лишається.

Відкритих даних щодо широти використання фреймворків немає, проте непрямої аналіз відкритих вакансій на ресурсі Indeed [34] дає досить цікавий розподіл:

- 1 місце – React 78,1%;
- 2 місце – Angular 21%;
- 3 місце – Vue.js 0,8%;
- 4 місце – без уточнення середовища 0,1%.

Дані опитування StackOverflow [35] надають вже дещо інші цифри: 67% голосів отримав React, тоді як за Angular «проголосувало» 42% програмістів.

Виходить, що найбільш популярний серед усіх – React. Але чи означає, що він буде ефективніше? Зовсім ні, адже за популярністю криється лише використання середовища в таких проектах, як Facebook, Instagram. До того ж, наприклад, на Github саме Vue.js має найбільше «зірочок» [36].

Тому варто відмовитися від чужого досвіду і спробувати порівняти JavaScript-фреймворки за фактичними показниками, які критичні для розробника.

Перший показник – *рендеринг*, відображення кінцевого результату. Javascript фреймворки в першу чергу варто порівняти відносно рендерингу сторінки. Сучасна архітектура допускає два види: на стороні клієнта (сторінка промальовується за рахунок потужностей ПК користувача) або на стороні сервера. Об'єктна модель документа DOM, що дозволяє зчитувати і змінювати вміст, оформлення і навіть структуру html-документів. Кожен з фреймворків підходить по-своєму до обробки DOM що і впливає на рендеринг кінцевої сторінки, яка відображається на екрані користувача. Vue.js і React створюють копію DOM, обробляють її, а потім результат порівнюється з вихідною версією. В кінцевому документі (тобто на екрані користувача) замінюються тільки ті частини сторінки, які відрізняються від результатів обробки. Це значно прискорює завантаження і рендеринг сторінки. Відповідно скорочується обсяг трафіку, що особливо важливо для користувачів мобільних пристроїв. У корені відрізняється підхід до обробки DOM фреймворком Angular версії 1.x і вище. Тут відбувається поділ на два потоки, причому за рендеринг DOM «відповідає» браузер (клієнтська частина), а за створення директив, завантаження коду і сервісів – загальний потік (серверна частина) [33]. Але це зовсім не означає, що рендеринг відбувається на стороні клієнта – візуалізація як і раніше

проводиться серверами. Отже, SEO оптимізація не викличе труднощів. Пошуковим роботам буде надана коректна сторінка при індексації.

Другий показник – *архітектура компонентів*. React не відноситься до фреймворків у чистому вигляді. Це свого роду модифікована бібліотека, «заточена» під MVC (Model-View-Controller, де Модель відповідає за надання даних, Вид – відображає дані Моделі користувачеві, а Контролер інтерпретує дії користувача і змушує Модель вносити зміни). Angular і Vue.js відносяться вже до самих фреймворків [33].

Якщо в основі архітектури проекту лежить React, то це зумовлює:

- необхідність пошуку та впровадження додаткових бібліотек для реалізації кожної з задач;
- налаштування функціональної частини програми під конкретну бібліотеку;
- складність залучення до проекту інших розробників, через різницю в стеці бібліотек кожної програми [33].

Отже, для реалізації архітектури знадобиться більше часу. Низькорівневий API вимагає занадто довгого налаштування. У разі використання готових фреймворків – Vue.js і Angular, проблем з підбором або налаштуванням бібліотек для різних завдань вже не виникає. Високорівнева API забезпечує зворотну сумісність для всіх бібліотек. Це дозволить підключитися до проекту сторонньому програмісту без тривалого вивчення архітектури додатку. Саме в уніфікації процесів криється популярність повноцінних фреймворків [33].

Третій показник – *спрямованість і класи залежностей*. React і Vue.js підтримують тільки односторонню передачу даних. При цьому в React об'єкти інкапсульовані. Говорячи простою мовою, кожен з об'єктів програми відноситься до кінцевих процедур, які не вимагають дій користувача до закінчення роботи. Однак React підтримує копіювання та передачу стану. Тобто, властивості прописаних об'єктів можуть бути відновлені на іншому пристрої, якщо запустити застосування і повідомити стан компонентів. Отже, рендеринг буде ідентичним, на екранах обох пристроїв буде одна і та сама «картинка». Фреймворк Vue.js працює вже трохи інакше [33]. JavaScript як і раніше односторонній, але компоненти працюють з шаблонами, і

на виході виходить чистий html. Є підтримка JSX, що спрощує перехід з React і схожих бібліотек. Для фреймворка автоматизована логіка процесів за рахунок побудови директив. Достатньо додати потрібну директиву, щоб отримати додаткову функціональність. Наприклад, необхідно зробити блоковий елемент переміщуються по сторінці. Для реалізації достатньо додати директиву «draggable» до властивостей об'єкта, після чого його можна буде перетягувати мишкою по екрану. Фреймворк Angular дещо відрізняється логікою процесу. Так, тут є все, що властиво Vue.js, однак опис взаємодії об'єктів відбувається в службах, які є складовими частинами модулів [33]. Модульна архітектура більш зручна під час розробки великих застосунків. Модуль створюється для вирішення декількох завдань схожої функціональності. Отже, кінцевий розмір коду стає меншим, а швидкість обробки вищою. Підтримка MVVM (Model-View-ViewModel) допускає вирішення різних задач в одному розділі програми з використанням одного набору даних. Залежність функцій визначає двосторонню спрямованість передачі даних. Кожна процедура може запустити інший процес [33].

Четвертий показник – *зворотня сумісність*. Для розробника додатку важлива можливість відновлення архітектури для підключення нових модулів і бібліотек. Angular – повноцінний фреймворк. Має абсолютну залежність від попередніх версій і компонентів. Прямий перехід з 4.0 на 5.0 неможливий, доведеться послідовно встановлювати всі оновлення між версіями. Це призведе до необґрунтованого збільшення обсягу програми. До речі, цікавий факт, Angular версії 3.0 не існує. Після 2-ї версії відразу йде 4-я. React – бібліотека. Повна сумісність між версіями. Можливе підключення до застосування бібліотек різних версій, оновлення застарілих зі спадкуванням властивостей. Vue.js – прогресивний фреймворк (згідно із заявою головного розробника Vue.js Евана Ю). Модульна система аналогічна React, але включає в себе всі атрибути JS-фреймворка, що працюють з повною зворотною сумісністю [33].

П'ятий показник – *техпідтримка*, документація і ком'юніті. При виборі якоїсь певної архітектури не можна ігнорувати підтримку спільноти програмістів (ком'юніті). Адже навіть прості функції можуть викликати труднощі. І чим складніше

проект, тим частіше розробник звертається і до технічної документації обраної середовища, і за порадою до ком'юніті. Тому варто порівняти підтримку кожного JS-фреймворків. React має досить непросто організовану техпідтримку. З одного боку, ком'юніті складають тисячі програмістів в різних країнах, що повністю знімає мовний бар'єр. Порадитися можна будь-якою мовою – українською, російською, англійською, китайською та іншими. Однак відкритий вихідний код і щоденна поява нових бібліотек на базі React є недоліками середовища. Складно знайти документацію, яка б вичерпно описувала процеси застосування, функції та властивості конкретної свіжої випущеної бібліотеки. Найчастіше нові модулі з'являються таким чином: програміст створює бібліотеку для вирішення якогось завдання в своєму проекті. Потім він пише інструкцію і викладає напрацювання в Інтернет. Vue.js має детальну документацію, що є візитною картою для фреймворка. Однак прихильники середовища не згадують, що велика частина інструкцій не має нормального перекладу ні англійською, ні російською мовою. Спільнота користувачів, незважаючи на велику популярність Vue.js на Github, доволі мала, особливо в порівнянні з Angular або React. Навіть JQuery користується набагато більше розробників, ніж Vue.js. Angular, на відміну від Vue.js і React, має не тільки детально написану документацію, але також для фреймворка випущено чимало гайдлайнів. Отже, не доведеться вдаватися до допомоги спільноти для пошуку відповідей. Кількість програмістів, що використовують фреймворк, обчислюється сотнями тисяч. Розробник в будь-який момент може поставити запитання з розряду «Howto». Для цього не доведеться шукати офіційний сайт проекту. Обов'язково знайдеться людина, яка знає, як вирішити поставлену задачу або підкаже, де знайти відповідь.

Враховуючи ці п'ять наведених показників, можна зробити певні висновки щодо фреймворків, які розглядаються у даному розділі:

- якщо потрібно швидко вивчити середовище, то варто вибирати між Vue.js і React;
- на Vue.js легко перейти як користувачу Angular, так і React. Адже тут виходить чистий html-код, знайомий всім розробникам. Прийоми і техніки використовуються приблизно ті самі, що і в Angular;



- якщо передбачається розробка великого проекту, то варто розглядати Angular в якості основи. Він забезпечує максимальну гнучкість і швидкість рендеринга. Величезний досвід інших розробників дозволить вирішити питання, які обов'язково виникнуть при роботі над застосуванням. React виявиться занадто об'ємним, а для Vue.js ще не існує великої кількості гайд-лайнів;
- якщо до розробки в майбутньому будуть залучатися інші програмісти, то Vue.js стане кращим вибором. Адже цей фреймворк не тільки простий для вивчення, а й дозволяє міняти додаток без руйнування його архітектури;
- якщо для проекту передбачається багатоступінчате оновлення і розширення функціональності в майбутньому, то варто використовувати Vue.js або React через чудову зворотню сумісність.

Враховуючи те, що часу на реалізацію програмного продукту не так багато, і вже є наявні певні знання бібліотеки React, вирішено використовувати саме її для розробки даного проекту.

### **3.1.4 Вибір середовища розробки**

Під час програмування важливим елементом є середовище розробки. Для проекту, який розробляється, буде використовуватись наступний продукт – JetBrains WebStorm [37]. Це IDE [38] або інтегроване середовище для розробки на мові JavaScript, яке підходить для client-side-розробки, створення застосувань на Node.js і мобільних застосувань на React Native. Головне достоїнство WebStorm – це зручний і розумний редактор для JavaScript, HTML і CSS, який також підтримує TypeScript, CoffeeScript, Dart, Less, Sass і Stylus і фреймворки, наприклад, Angular, React і Vue.js [37].

WebStorm, як і інші IDE на платформі IntelliJ IDEA [39], робить розробку простішою та зручнішою. WebStorm забезпечуючи підсвічування та автодоповнення коду, перевіряє його на помилки, допомагає швидко навігироватись за проектом і безпечно вносити зміни за допомогою рефакторингов. У WebStorm є інструменти для налагодження коду і інтеграція з системами управління версіями [37].

WebStorm по-справжньому розуміє структуру проекту і код, виявляє можливі проблеми ще до того, як розробник відкриє проект в браузері, і пропонує їх вирішення. Вбудовані в IDE інструменти для тестування допоможуть в розробці і зроблять її зручнішою і продуктивнішою.

Це середовище розробки, яке має дуже велику функціональність. Зручність полягає в тому, що все зібрано в одному місці. Безумовно, деякі ключові функції варті виділення і трохи більш докладного опису.

Перша особливість – розумна допомога при написанні коду. WebStorm допомагає писати код на JavaScript, використовуючи мови, що компілюються в JavaScript, HTML і CSS, а також з використанням Node.js. Автодоповнення коду, зручна навігація, миттєве виявлення помилок і рефакторінг доступні для всіх цих мов

*Сучасні фреймворки.* WebStorm допомагає розробляти програму для використання Angular, React, Vue.js і Meteor. Якщо ви використовуєте React Native, PhoneGap, Cordova або Ionic для мобільної розробки або Node.js, WebStorm теж зможе вам допомогти працювати продуктивніше. І все це в одній IDE.

*Розумний редактор.* На основі аналізу коду в проекті IDE пропонує найбільш релевантні варіанти автодоповнення для всіх підтримуваних мов. Сотні вбудованих інспекцій миттєво вказують на можливі помилки та пропонують варіанти їхнього виправлення.

*Навігація та пошук.* WebStorm допомагає швидко переміщатися по коду і економити час при роботі над великими проектами. В один клік можна перейти до визначення будь-якого методу, функції або змінної або знайти їхнє використання в проекті.

Друга особливість – налагодження, трасування та тестування. WebStorm має вбудовані інструменти для налагодження, трасування та тестування клієнтських і Node.js-застосувань. Продумана інтеграція дозволяє вирішувати ці завдання набагато простіше і без складних налаштувань.

*Налагодження.* У WebStorm вбудовано інструмент для налагодження клієнтського коду (працює в браузері Chrome) і застосувань на Node.js. Прямо в IDE

можна ставити точки зупинки, здійснювати послідовне виконання коду і перевіряти змінні.

*Тестування.* Тестувати програми за допомогою WebStorm простіше, так як IDE інтегрується з Karma, Mocha, Jest і Protractor. Можна запускати і налагоджувати тести прямо в IDE, переглядати результати в зручному форматі і легко переходити до тестованого коду.

*Трасування і профілювання.* Вбудований інструмент spy-js допомагає трасувати JavaScript-код. Spy-js покаже, як файли пов'язані з викликами функцій, і виявить фрагменти коду, виконання яких зайняло особливо багато часу.

Третя особливість – тісна інтеграція з інструментами. WebStorm підтримує популярні інструменти для веб-розробки, дозволяючи користуватися ними більш ефективно прямо в IDE, без командного рядка.

*Інструменти збірки.* Простий універсальний інтерфейс дозволяє запускати завдання за допомогою Grunt, Gulp і npm прямо з IDE. Всі завдання відображаються в спеціальному вікні і запускаються подвійним кліком на назву завдання.

*Інструменти для аналізу якості коду.* На додаток до сотень власних інспекцій WebStorm може перевіряти код за допомогою ESLint, JSCS, TSLint, Stylelint, JSHint або JSLint і підсвічувати помилки на льоту прямо в редакторі.

*Шаблони проектів.* Можна створювати нові проекти прямо зі стартового екрану WebStorm, використовуючи популярні шаблони, такі як Express або Web Starter Kit, і отримати доступ до безлічі інших генераторів проектів завдяки інтеграції з Yeoman.

Четверта особливість – сама функціональність IDE. WebStorm побудована на базі платформи IntelliJ, яку в JetBrains розвивають і удосконалюють вже більше 15 років. Продумана стандартна конфігурація в поєднанні з широкими можливостями налаштування IDE роблять WebStorm зручним середовищем для всіх розробників.

*Підтримка систем контролю версій.* У WebStorm реалізований універсальний інтерфейс для роботи з популярними системами контролю версій, тому працювати з проектами, які використовують Git, SVN, Mercurial і Perforce, однаково зручно.

*Локальна історія.* Незалежно від того, користується розробник системою контролю версій чи ні, іноді саме локальна історія змін може врятувати проект.

Можна в будь-який момент переглянути історію змін окремого файлу або папки і відкотитися до будь-якої з попередніх версій.

*Можливості індивідуального налаштування.* WebStorm дозволяє повністю налаштувати робоче середовище. Можна адаптувати IDE під свої потреби: вибрати зручні комбінації клавіш, тему оформлення та розташування елементів редактора.

Описавши можливості даного середовища розробки, також варто вказати ще одну його перевагу – воно безкоштовно надається для студентів вищих навчальних закладів з усього світу. Достатньо тільки підтвердити цей статус, для цього існує декілька способів, найдовший з них, як заявляють саме на сайті продукту – до тижня, але насправді доступ був отриманий вже за добу. Надсилається інформація про вищий навчальний заклад та скан-копія студентського квитка. Після підтвердження на електронну пошту прийде лист-інструкція щодо активації студентської ліцензії на один календарний рік.

Варто також зазначити, що інтерфейс є дуже зручним. Можливість налаштування дрібних деталей, «м'які» кольорові поєднання для ока дозволяють не втомлюватися під час тривалої роботи над певним проектом. Приклад вигляду інтерфейсу IDE зображений на рисунку 3.1 нижче.

Як видно з ілюстрації, інтерфейс виглядає дуже простим і ненагромадженим, кожен елемент за замовчанням займає зручне місце, але все можна також налаштувати «під себе». Також посередині можна побачити впливаюче вікно з підказками, яке демонструє частину функціоналу середовища для розробки WebStorm.

Також варто зазначити, що даний продукт є кросплатформним – ПЗ доступне для операційних систем Windows, MacOS, Linux. Проекти мають єдину архітектуру, тобто почавши роботу в одній операційній системі, можна цей проект відкрити в іншій операційній системі, що, безумовно, є дуже зручним

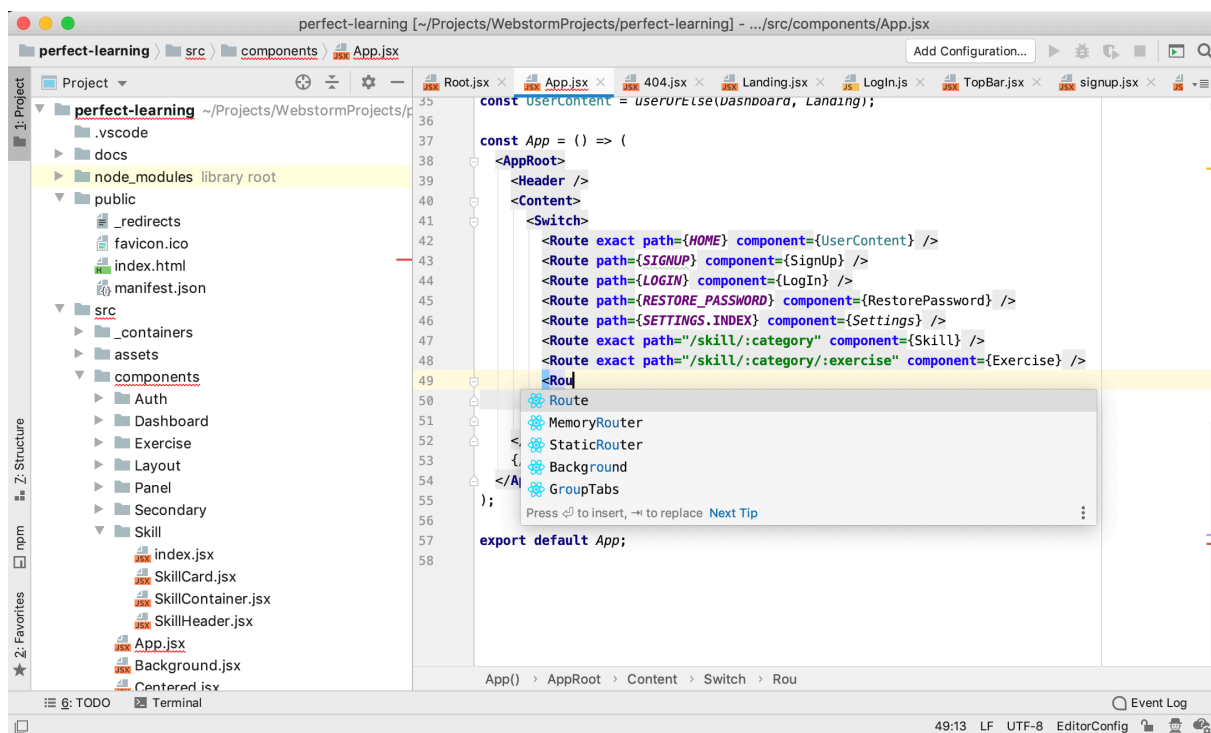


Рисунок 3.1 – Інтерфейс WebStorm IDE

Отже, враховуючи всі описані переваги, було вирішено використовувати дане середовище для розробки під час роботи над програмним продуктом.

### 3.1.5 Вибір сервісу для реалізації backend-складової системи

Оскільки застосування працює з багатьма людьми, то в роботі оговорювалось, що потрібно щось схоже на клієнт-серверну архітектуру. Якщо з першим проблем нема, і вже вирішено, що це буде веб-застосування на React, то потрібно розуміти, де будуть оброблятися і зберігатися дані. Тут на допомогу приходить сервіс Firebase [40], який досить класно поєднується технологічно з бібліотекою React.

Firebase – це BaaS (Backend-as-a-Service) – рішення, яке пропонує серверні рішення у якості сервісу. Як мінімум, це SDK, сховище і база даних у хмарі, які є абсолютно безкоштовними.

Firebase допомагає швидко створювати якісні програми, збільшувати аудиторію залучених користувачів і підвищувати доходи. Платформа містить безліч корисних функцій для застосування, в тому числі серверний код для мобільних сервісів, статистику, а також інструменти для монетизації і розширення аудиторії.

*Навіщо це потрібно?* Пакет розробника Firebase об'єднує інтуїтивно зрозумілі API, позбавляючи розробника від необхідності керувати окремими пакетами. Можна

вибрати тільки те, що потрібно, і користуватися перевагами інтегрованого рішення. Платформа, яка використовує інфраструктуру Google, надає необхідні можливості для кожного етапу розробки і зростання [41].

Головні переваги даного рішення описані нижче.

*Швидкість роботи.* У пакеті розробника Firebase зібрані інтуїтивно зрозумілі API, які спрощують і прискорюють розробку якісних застосунків. Також у вашому розпорядженні всі необхідні інструменти для розширення користувацької бази і підвищення доходів – розробнику залишається тільки вибрати відповідні для ваших цілей [41].

*Готова інфраструктура.* Розробнику не доведеться створювати складну інфраструктуру або працювати з декількома панелями управління. Замість цього він зможе зосередитися на потребах користувачів [41].

*Статистика.* В основі Firebase лежить безкоштовний аналітичний інструмент, розроблений спеціально для мобільних пристроїв. Google Analytics для Firebase дозволяє отримувати дані про дії користувачів і відразу ж вживати заходи за допомогою додаткових функцій [41].

*Кросплатформеність.* Firebase працює на будь-яких платформах завдяки пакетам розробника для Android, iOS, JavaScript і C ++. Ви також можете звертатися до Firebase, використовуючи серверні бібліотеки або REST API [41].

*Масштабованість.* Якщо застосування стане популярним і навантаження на нього зросте, вам не доведеться міняти код сервера або залучати додаткові ресурси – Firebase зробить це самостійно. Крім того, більшість функцій Firebase безкоштовні і залишаються такими незалежно від масштабу ваших проектів. Платних функцій чотири. У них передбачено безкоштовний пробний період і два тарифні плани [41].

*Безкоштовна підтримка електронною поштою.* Крім того, команда Firebase і фахівці з розробки Google дадуть відповіді на ваші питання на ресурсах Stack Overflow і GitHub.

Отже, використання даного сервісу дозволить значно полегшити і прискорити процес розробки майбутньої системи.

Тому, пара засобів для реалізації веб-застосування виглядає наступним чином – React і Firebase.

### **3.2 Вимоги до технічного забезпечення**

Визначившись із засобами розробки, необхідно також визначити цільові пристрої, на яких можна буде використовувати систему. У сучасних реаліях складно знайти комп'ютер чи телефон, якому буде більше 7-10 років, але такі бувають. Головне, щоб їхня конфігурація мала мінімально необхідний на сьогодні поріг, який допустимий для запуску операційної системи, користування веб-браузером тощо.

Для коректного та швидкого функціонування розробленої системи у складі технічних засобів для розробленої системи має бути комп'ютер, який має наступні параметри:

- процесор з тактовою частотою не нижче 1.1 ГГц;
- об'єм оперативної пам'яті RAM не менше 2048 МБ;
- ємність жорсткого диску HDD або твердотілого накопичувача SSD не нижча за 20 ГБ;
- постійний доступ до мережі Інтернет.

Для роботи клієнтського застосування повинні використовуватись наступні периферійні пристрої, під'єднані до комп'ютера:

- монітор;
- клавіатура дротова або бездротова;
- мишка або трекпад.

Також варто зазначити, що запуск застосування можливий також на планшетах та смартфонах, які теж відповідають мінімальним технічним вимогам сьогодення і мають доступ до мережі Інтернет.

Окрім цього, для запуску веб-застосування повинно бути встановлене додаткове програмне забезпечення, а саме – один з перелічених веб-браузерів:

- Google Chrome версії 45 або більш новий;
- Microsoft Edge версії 20 або більш новий;
- Mozilla Firefox версії 43 або більш новий;
- Opera версії 30 або більш новий;

– Safari версії 12 або більш новий.

У таких браузерах, як, наприклад, Internet Explorer та Opera Mini, можливі збої у роботі програми. Не рекомендується використовувати дані браузери для запуску застосування.

Версії браузерів вказані навмисно, оскільки машина може бути трохи застарілою, і навіть вже не підтримувати деякі системні оновлення, але при цьому її потужностей ще достатньо, щоб мати можливість встановлювати актуальні оновлення браузерів і іншого програмного забезпечення. Оскільки інструменти для програмування постійно з'являються нові, оновлюються, розширюють свій функціонал, то зрозуміло, що старе програмне забезпечення вони не будуть підтримувати.

### **3.3 Архітектура програмного забезпечення**

Архітектура для програмного забезпечення була визначена у розділах 1.3 та 3.1 даної роботи. Було вирішено, що воно буде реалізовано у вигляді веб-застосування на базі React. Для реалізації серверної частини буде використовуватись сервіс Firebase, який і надає всі ці серверні послуги в хмарі, так само як і є провайдером бази даних, яка буде документо-орієнтованою. У розділі 1.3 була описана її ER-модель.

У даному розділі буде розглядатися система у цілому, її компоненти та схеми послідовності.

#### **3.3.1 Діаграма послідовності**

Дана діаграма дозволяє продемонструвати високорівневі відношення між акторами, системою і сховищем для даних.

Після проходження студентом тесту системою генерується статистичний звіт. Студент має змогу переглянути цю статистику. У той самий час вона надсилається до сховища даних.

Викладач має можливість переглядати статистичну інформацію одразу зі сховища або використовуючи спеціальну панель для адміністративного керування. Переглядаючи й аналізуючи статистику, викладач має змогу змінювати тести або самі питання, які наявні в його дисципліні.



Схема структурна послідовності для системи, що розглядається, наведена на рисунку 3.2 нижче.

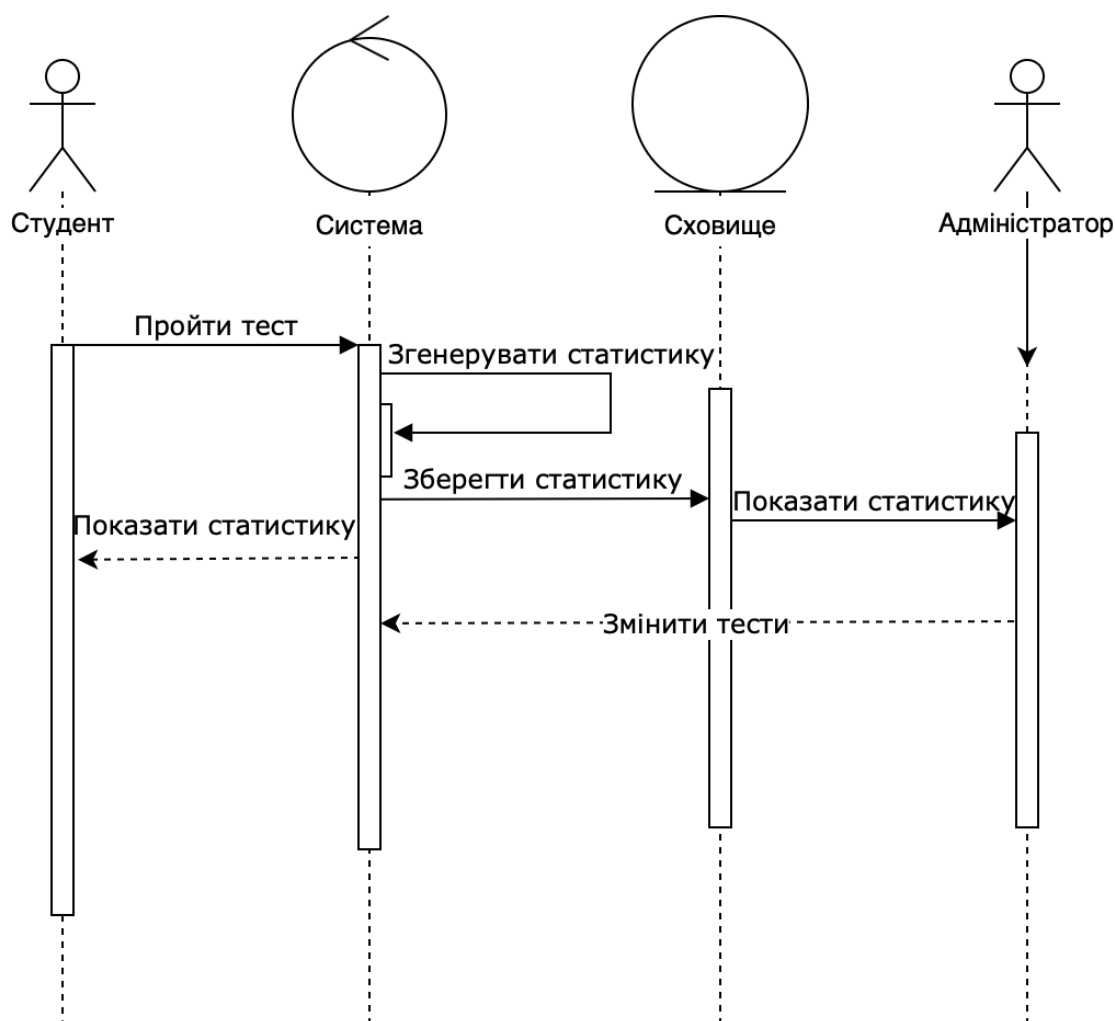


Рисунок 3.2 – Схема структурна послідовності

### 3.3.2 Діаграма компонентів

Діаграма компонентів дозволяє продемонструвати основні компоненти всередині системи. Нижче наведене пояснення зв'язків між ними.

- компоненти-подання отримують стан через їхню функціональну композицію із компонентами-контейнерами;
- контейнери мають змогу безпосередньо взаємодіяти із зовнішніми сервісами для отримання та/або відправлення даних;
- контейнери мають безпосередню взаємодію із даними, що стосуються самостійних тестів;
- контейнери мають змогу відправляти глобальні дії, реакція на які в майбутньому дозволить змінити глобальний стан системи;

- відбувається асинхронна взаємодія між клієнтом і зовнішніми сервісами;
- інформація, отримана від зовнішніх сервісів, надходить до сховища та змінює стан;
- інформація зі сховища надходить до контейнерів для перетворення і подальшої передачі у подання.

Схема структурна компонентів для системи, що розробляється наведена на рисунку 3.3 нижче.

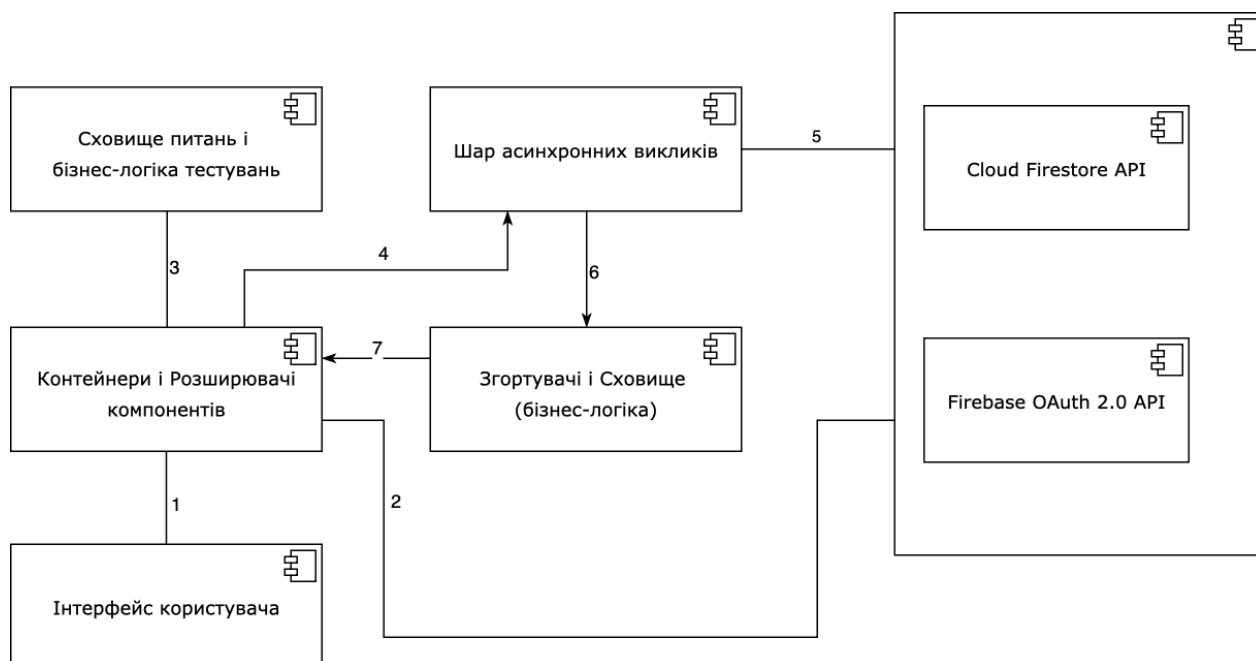


Рисунок 3.3 – Схема структурна компонентів

### 3.4 Настанова користувача

*Підготовка до роботи.* Для того, щоб почати роботу із застосуванням, потрібно у першу чергу запустити веб-браузер на своєму пристрої. Після цього, необхідно перейти за вказаною адресою, щоб отримати доступ до веб-застосування. Передбачено, що кожен навчальний заклад сам вирішуватиме, де розмістити систему, а отже, сам обере адресу. Тому, адресу розміщення системи дистанційного навчання (СДН) необхідно запитати безпосередньо у відповідальних за неї у навчальному закладі. Також, доволі часто на головному сайті закладу можна побачити посилання на СДН, і одразу почати роботу.

*Початок роботи.* Після першого запуску, або якщо користувач не був зареєстрованим або авторизованим у системі, застосування видає початкову сторінку, яка зображена на рисунку 3.4 нижче.

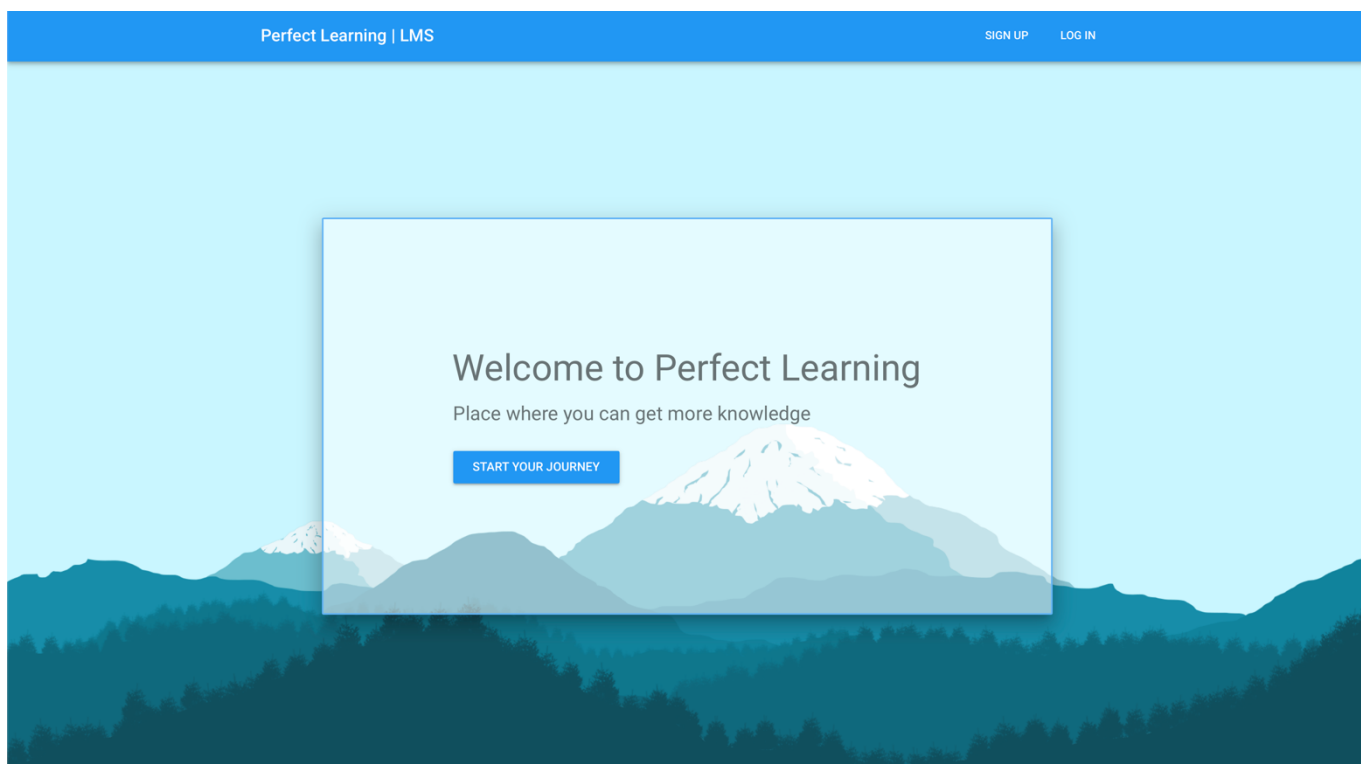


Рисунок 3.4 – Початкова сторінка застосування

Дана сторінка відображається для усіх – студент або вчитель, не має значення, оскільки користувач ще не пройшов аутентифікацію. Як видно, зверху є також меню, яке дозволяє або пройти до реєстрації, або до авторизації. Також кнопка в центрі сторінки теж веде до сторінки авторизації, що полегшує взаємодію із користувачами.

*Авторизація/Реєстрація.* Користувач має дві опції для авторизації або реєстрації. Перша – стандартна, за допомогою електронної пошти. Друга – за допомогою облікового запису Google, що можливо завдяки інтеграції з Firebase. Після проходження автентифікації користувач переходить на сторінку вибору доступних курсів.

*Організація робочого процесу.* Обравши потрібний курс, студент одразу бачить перед очима відкриті і закриті тести. Відкриті тести можуть бути доступні вперше або якщо студенту необхідно їх повторити через низьку запам'ятовуваність матеріалу, які вони покривають. Приклад сторінки з дисципліною продемонстровано на рисунку 3.5 нижче.

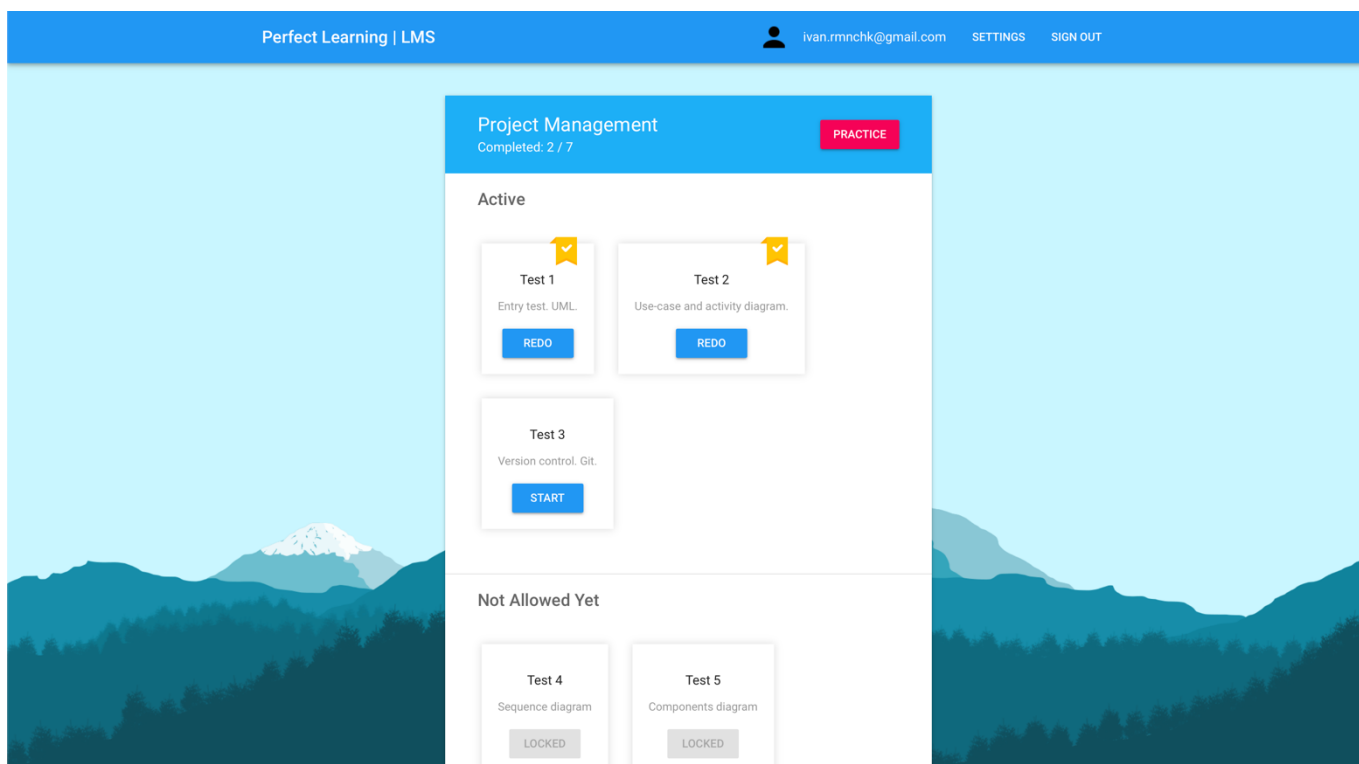


Рисунок 3.5 – Сторінка дисципліни з тестами

Функціонал доволі простий. Якщо тест новий – то на ньому зображена кнопка з написом “START”, якщо ж повторний – “REDO”. Тести, які наразі недоступні для студента, знаходяться нижче, їх неможливо розпочати. Вони стануть доступні або коли їх вперше відкриє викладач, або коли настане час для повтору.

*Переривання та завершення роботи.* У застосуванні передбачено переривання тестування, з попередженням, що після цього прогрес буде втрачено. Це просто додає користувачеві можливість відмовитись від розпочатого тесту, не закриваючи при цьому саме застосування. Щоб завершити роботу із застосуванням, необов’язково виходити з облікового запису. В такому випадку, сесія збережеться і не потрібно буде наступного разу повторно авторизовуватись. Якщо ви користуєтесь не особистою технікою, а публічною, то виходити з облікового запису обов’язково, щоб наступна людина не отримала до нього доступ і не нашкодила вашим статистичним або особистим даним. Після цього вкладку/вікно або сам браузер можна закрити.

### 3.5 Випробування програмного продукту

Кожен програмний продукт проходить випробування. Це необхідний етап контролю якості та відповідності функціональності програмного забезпечення. Тому, проводиться декілька контрольних тестів для основних функцій.

### 3.5.1 Загальні положення

Випробування програмного продукту регламентуються наступними нормативними документами:

- ГОСТ 34.603–92. Інформаційна технологія. Види випробувань автоматизованих систем;
- ГОСТ РД 50-34.698-90. Автоматизовані системи вимог до змісту документів.

### 3.5.2 Результати випробувань

Під час проходження випробувань була перевірена вся функціональність системи дистанційного навчання, яка забезпечує підтримку методики навчального процесу, заснованої на методі інтервальних повторень. Назви, опис і результати основних функцій наведемо у наступних таблицях 3.1-3.x, які наведені нижчі. Всі тестові випробування були виконані відповідно до регламенту.

Таблиця 3.1 – Перевірка функції “Авторизація”

Мета тесту	Перевірка функції “Авторизація”
Початковий стан системи	Відкрита сторінка авторизації (шлях ‘/login’)
Вхідні дані	Електронна адреса та пароль зареєстрованого користувача
Схема проведення тесту	Ввести електронну адресу і пароль у відповідні текстові поля форми, після цього натиснути на кнопку “Login”
Очікуваний результат	Відбувся перехід на головну сторінку
Наявний результат	Відбувся перехід на головну сторінку

Таблиця 3.2 – Перевірка функції “Перевірка коректності введених даних у процесі авторизації”

Мета тесту	Перевірка функції “Перевірка коректності введених даних у процесі авторизації”
Початковий стан системи	Відкрита сторінка авторизації (шлях ‘/login’)
Вхідні дані	Пошта: qwert435, пароль: 000
Схема проведення тесту	У поле поштової адреси ввести “qwert345” та пароль “000”, після чого натиснути на кнопку “Login”

Мета тесту	Перевірка функції “Дострокове завершення тесту”
Очікуваний результат	Отримати повідомлення/сповіщення про помилки.
Наявний результат	Отримано повідомлення про наступні помилки: “Invalid email” і “Password contains at least 8 characters”

Таблиця 3.3 – Повне виконання тесту

Мета тесту	Перевірка функції “Повне виконання тесту”
Початковий стан системи	Користувача авторизовано, вибрано один тест
Вхідні дані	- (відсутні)
Схема проведення тесту	Розпочати виконання обраного тесту. Продивитися тестові питання. Вибрати всі варіанти відповідей. Повністю пройти тест, без дострокового переривання.
Очікуваний результат	Перехід на сторінку зі статистикою з тесту
Наявний результат	Відбувся перехід на сторінку зі статистикою з тесту

Таблиця 3.4 – Дострокове завершення тесту

Мета тесту	Перевірка функції “Дострокове завершення тесту”
Початковий стан Системи	Користувача авторизовано, вибрано один тест
Вхідні дані	- (відсутні)
Схема проведення тесту	Розпочати проходження обраного тесту. Натиснути кнопку “End Test”. В активованій сторінці підтвердження завершення натиснути кнопку “Terminate”.
Очікуваний результат	Перехід на сторінку зі статистикою з тесту
Наявний результат	Відбувся перехід на сторінку зі статистикою з тесту

Отже, як видно з таблиць вище, випробування пройшли успішно. Це говорить про те, що продукт відповідає вимогам і гарантує якість.

### **Висновок до розділу**

У даному розділі в процесі роботи було розглянуто важливі питання, які стосуються створення та реалізації програмного продукту.

У ході роботи було визначено засоби розробки для майбутнього програмного забезпечення, а саме обрані цільова платформа, бібліотеки та сервіси для веб-застосування, середовище розробки.

Також тут була описана архітектура програмного забезпечення з наведенням діаграм компонентів і послідовності.

В розділі також було наведено керівництво для користувача, яке описує основні функції та надає рекомендації щодо роботи із веб-застосуванням.

Також для системи були проведені відповідні програмні випробування, тести основних функцій були наведені в даному розділі.

В результаті роботи досягнуто наступне: програмний продукт успішно пройшов випробувальну програму і готовий до запуску.

## 4 РОЗРОБКА СТАРТАП-ПРОЕКТУ

### 4.1 Опис ідеї проекту

Запропоновано систему підтримки методики навчання, заснованої на методі інтервальних повторень зі створенням індивідуального плану навчання та урахуванням особистих навичок студентів.

Відміною рисою від існуючих систем є те, що в основі системи лежить система дистанційного навчання, яка має додатковий функціонал – а саме, формує час та склад самостійних тестів, які мають бути пройдені студентами. Все це відбувається з урахуванням індивідуальних особливостей кожного з них.

Зібрані статистичні дані передаються через мережу Інтернет на сервер, де інформація зберігається. За допомогою цієї статистики можна зробити аналіз та висновки щодо того, наскільки добре засвоюється матеріал студентами у середньому, та яка його частина потребує додаткової уваги через низькі показники.

*Призначенням* цього комплексу є підвищення рівня запам'ятовуваності навчального матеріалу студентами.

*Сутність розробки* створення системи дистанційного навчання із клієнт-серверною архітектурою полягає у підвищенні ефективності навчання за рахунок підняття рівня запам'ятовуваності матеріалу студентами.

*Цільовою аудиторією* є навчальні заклади, які зможуть використовувати дану систему у навчальному процесі.

*Основними вигодами використання* є підвищення запам'ятовуваності навчального матеріалу студентами, що в свою чергу підвищує якість навчання та рейтинг навчального закладу з меншими витратами.

### 4.2 Опис конкурентів

У процесі збору інформації про аналоги було знайдено декілька закордонних проектів. Докладний опис аналогів зроблений у розділі 1.3 даної роботи.

До закордонних проектів належать наступні системи дистанційного навчання: *ATutor, Claroline, Live@Edu, eFront, SharepointLMS, Moodle*. Усі наведені системи, окрім останньої у списку використовуються навчальними закладами у всьому світі. Але в Україні вони не є поширені, тому не можуть розглядатися з конкурентної



позиції. Також до аналогів можна віднести два інші проекти – навчальний сервіс програмування *Codecademy* та сервіс вивчення іноземних мов *Duolingo*. Останній трохи відрізняється сферою діяльності, але має всередині програмний алгоритм, який базується на тих самих основах, на яких базується розроблений алгоритм для цієї системи. При цьому, він не є ближчим аналогом, бо стосується вивчення іноземних мов – тобто, цільова аудиторія трохи відрізняється.

Ближчим аналогом до поточної системи є також закордонний проект, згаданий вище у переліку систем дистанційного навчання (СДН) – *Moodle*. Це СДН, яка є дуже популярною серед вищих навчальних закладів в Україні і застосовується сьогодні.

Зовнішній вигляд системи, що надається користувачеві, зображений на рисунку 4.1.

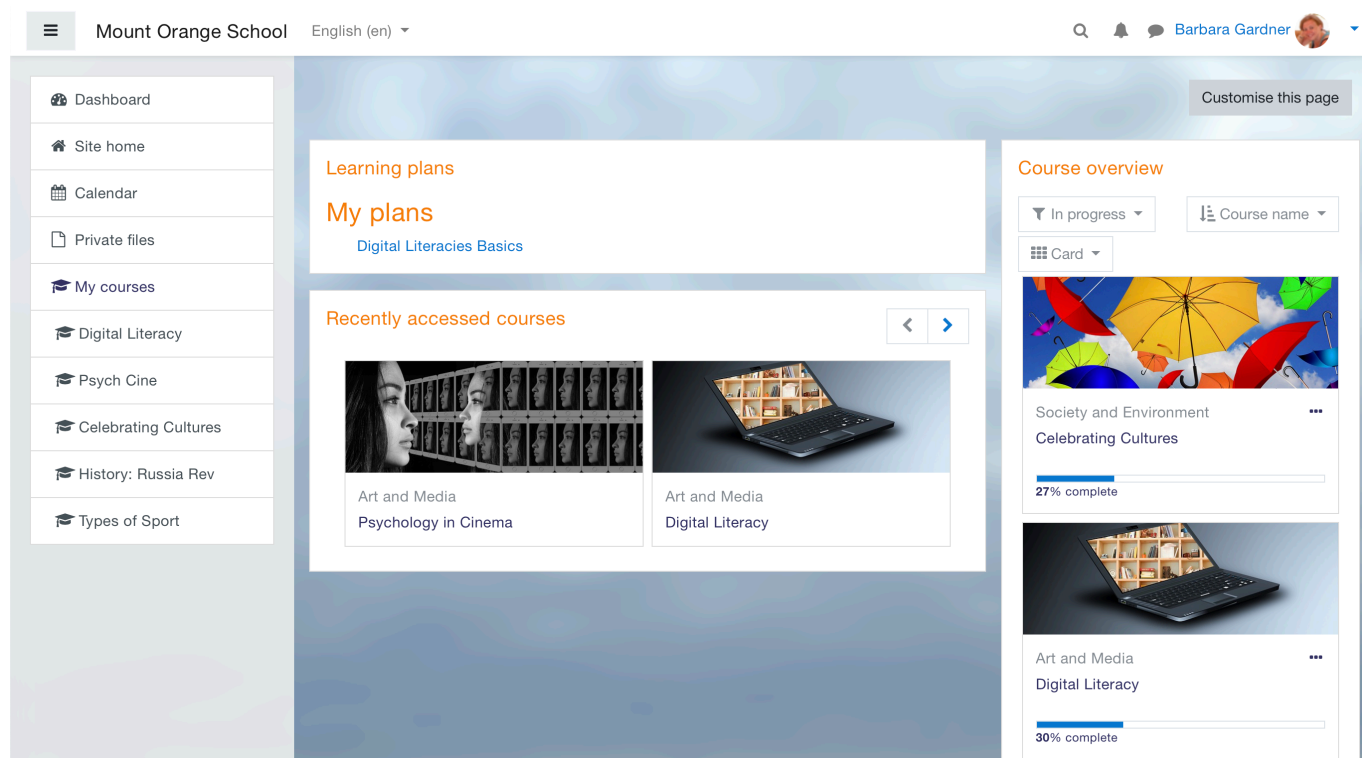


Рисунок 4.1 – Інтерфейс сторінки студента в СДН Moodle

Згідно зі сторінкою інформації про систему, Moodle є навчальною платформою, яка використовується у багатьох країнах світу. Вона є безкоштовною. Також система має мережу з більше 80 партнерських компаній по всьому світу, які забезпечують підтримку у різних країнах. В Україні також є офіційний партнер Moodle – *Technomatix* [42].

Інша опція, яку пропонує сервіс – це надання доступу до хостингу з різними тарифними планами. Тобто, вони розгортають ці системи на власному хостингу MoodleCloud. За цю опцію вже необхідно платити, оплата на основі щорічної підписки. Тарифна сітка – щорічний платіж 0-1000 AUD (австралійських доларів) [43].

Роблячи висновки щодо конкуренції, то певні ризики є, оскільки наразі цільова аудиторія даного сервісу збігається з цільвою аудиторією моєї системи, а отже, доведеться за неї поборотися. Систему можна назвати головним конкурентом, але не сильною загрозою. Також її можна взяти до порівняння через схожий функціонал.

Порівняння двох проектів розпочнемо із виокремлення загальної ідеї та зведення їх у вигляді таблиць.

Сильні, слабкі та нейтральні сторони проекту згруповані та наведені у таблиці 4.1, разом із даними про конкурента.

Таблиця 4.1 – Визначення сильних, слабких та нейтральних характеристик ідеї проекту.

Ідея	(потенційні) товари/концепції конкурентів		Слабка сторона (W)	Нейтральна сторона (N)	Сильна сторона (S)
	<i>Мій проект</i>	<i>Moodle</i>			
<i>Навчальна платформа для студентів і викладачів</i>	Наявна	Наявна		+	
<i>Формування індивідуального плану тестування</i>	Наявне	Відсутнє			+
<i>Збір статистики для користувачів</i>	Наявний	Наявний		+	

Продовження таблиці 4.1

Ідея	(потенційні) товари/концепції конкурентів		Слабка сторона (W)	Нейтральна сторона (N)	Сильна сторона (S)
	<i>Мій проект</i>	<i>Moodle</i>			
<i>Відбір матеріалу для повторення</i>	Наявний	Відсутній			+
<i>Оцінювання студентів</i>	Наявне	Наявне		+	

На основі визначеного переліку сильних, слабких та нейтральних сторін ідеї можливо провести аналіз та зрозуміти його конкурентоспроможність. Як бачимо з таблиці 4.1, потенційна ідея має значні переваги до відношення до свого конкурента та може випередити його.

#### 4.3 Технологічний аудит ідеї проекту

Проведемо аналіз технічної здійсненності проекту, оцінивши можливі технології реалізації. У таблиці 4.2 наведено відношення ідей проекту з технологіями, які забезпечать їх виконання.

Таблиця 4.2 – Технологічна здійсненність ідеї проекту

Ідея проекту	Технології реалізації	Наявність технології	Доступність технології
<i>Навчальна платформа для студентів і викладачів</i>	Веб-застосування з використанням бібліотеки React	Наявна	Доступна
<i>Формування індивідуального плану тестування</i>	Визначення оптимального терміну тестування	Реалізується в системі	Доступна
<i>Збір статистики для користувачів</i>	Сервер на базі <i>Firebase</i>	Наявна	Доступна

<i>Відбір матеріалу для повторення</i>	Алгоритм визначення слабких місць за статистикою у базі даних	Реалізується в системі	Доступна
<i>Оцінювання студентів</i>	Обчислення та форматування статистичних даних	Наявна	Доступна

Наведемо список з наявних та обраних технологій для проекту та обґрунтуємо їх використання:

- *навчальна платформа для студентів і викладачів* – веб-застосування з використанням бібліотеки React;
- *формування індивідуального плану тестування* – визначення оптимального терміну тестування;
- *збір статистики для користувачів* – сервер на базі *Firebase*;
- *відбір матеріалу для повторення* – алгоритм визначення слабких місць за статистикою у базі даних;
- *оцінювання студентів* – обчислення та форматування статистичних даних.

Згідно з обраним списком технологій, можливо зробити висновок, що задумка проекту з огляду на технічну частину є цілком здійсненою.

#### **4.4 Аналіз ринкових можливостей запуску стартап-проекту**

Підчас впровадження стартап-проекту на ринок, в першу чергу, постають питання щодо ринкових можливостей проекту, та загроз, що можуть зашкодити підприємству.

Проведемо аналіз на наявність попиту та його обсяг. Для цього оцінимо головних гравців ринку та зведемо дані у таблиці 4.3 нижче.

Таблиця 4.3 – Попередня характеристика потенційного ринку стартап-проекту

Показник стану ринку	Характеристика
<i>Кількість головних гравців</i>	1. Moodle
<i>Загальний обсяг продажів</i>	Кількість користувачів системи: 174 000 000+ Кількість сайтів в Україні: 724 [44]
<i>Динаміка ринку</i>	Зростає
<i>Наявність обмежень для входу</i>	Стартовий капітал та написання програмного забезпечення
<i>Специфічні вимоги до стандартизації та сертифікації</i>	Обов'язкові вимоги відсутні на законодавчому рівні. Щодо логічних вимог – інтерфейс і функціональність.
<i>Середня норма рентабельності в галузі</i>	Невідома

**Наявність обмежень для входу.** Основою роботи стартап-проекту є розробка системи у вигляді веб-застосування для можливості розгортання в кожному окремому навчальному закладі.

**Специфічні вимоги до стандартизації та сертифікації.** Система дистанційного навчання, яка розглядається, не потребує обов'язкової сертифікації чи стандартизації. Вона повинна мати відповідний інтерфейс і функціонал, але на законодавчому рівні це не регулюється. Освітні ліцензії не потрібні, оскільки система не буде видавати сертифікати, які будуть рівносильні освітнім атестатам чи дипломам.

**Середня норма рентабельності в галузі.** Механізмом монетизації для компанії конкурента є розгортання системи на власному хостингу. Проведення оцінки обсягу продаж конкурента зробити неможливо, через відсутність відкритої аналітики монетизації.

Наведемо опис цільових груп проекту у таблиці 4.4 нижче.

Таблиця 4.4 – Характеристика потенційних клієнтів стартап-проекту

Потреба	Цільова аудиторія	Відмінності у поведінці цільових груп	Вимоги споживачів до товару/послуги
<i>Комфорт використання системи</i>	Студенти та викладачі	Відсутні	Простота та зрозумілість
			Стабільна робота в умовах одночасного підключення багатьох людей
			Правильне функціонування компонентів
<i>Оптимізація витрат на обслуговування маршруту</i>	Навчальний заклад	Відсутні	Надання даних щодо успішності студентів, пов'язаних з викладачем

Після визначення цільових груп споживачів послуг та їх вимог, логічним є аналіз ринкового середовища, а саме визнання факторів загроз та можливостей.

Відомою тактикою боротьби з загрозами є передбачення їх можливих джерел та планування роботи з уникання або зведення наслідків до прийнятних втрат. З відомих на поточний час фактів, виокремимо існуючі та майбутні загрози та наведемо шляхи їх подолання у таблиці 4.5.

Таблиця 4.5 – Фактори загроз

Фактор	Зміст загрози	Реакція компанії
<i>Кадри</i>	Необхідне навчання кадрів для встановлення та налаштування ПЗ	Пошук та навчання людей
<i>Відмова від співпраці</i>	Навчальний заклад відмовляється від співробітництва	Отримання коментарів щодо причини відмови, введення змін, пропонування компромісу

Продовження таблиці 4.5

<b>Фактор</b>	<b>Зміст загрози</b>	<b>Реакція компанії</b>
<i>Небажання користувачів платити за додатковий матеріал</i>	Неокупність проекту	Інтегрування реклами та розширення безкоштовного функціоналу.
<i>Недостатнє фінансування</i>	Недостатнє фінансування компаній	Аналіз та оптимізація витрат, маркетингові заходи з пошуку клієнтів

Визначимо фактори можливостей, які у змозі позитивно вплинути на стан компаній. Згруповані дані наведемо у таблиці 4.6.

Таблиця 4.6 – Фактори можливостей

<b>Фактор</b>	<b>Зміст можливості</b>	<b>Реакція компанії</b>
<i>Науково-технічний</i>	Покращення алгоритму підбору завдань	Впровадження в роботу
<i>Можливість встановлення монополії</i>	Можливість захопити ринок послуг	Робота над вдосконаленням системи
<i>Робота з закордонними замовниками</i>	Встановлення контактів з іноземними замовниками	Локалізація, інтернаціоналізація та сертифікація системи
<i>Економічний</i>	Підтримка інновацій у виробництві	Підвищення / Пониження ціни на послугу

Проаналізувавши ринкове середовище, можна зробити висновок щодо найбільшої загрози підприємству – недостатнє фінансування, тому проект повинен буде зосередити усі зусилля на пошук нових клієнтів та підтримку вже існуючих зв'язків.

Надалі проведемо аналіз пропозицій на ринку та визначимо загальні риси конкуренції на ринку у таблиці 4.7 нижче.

Таблиця 4.7 – Ступеневий аналіз конкуренції на ринку

<b>Особливості конкурентного середовища</b>	<b>В чому проявляється дана характеристика</b>	<b>Вплив на діяльність підприємства</b>
<i>Монополія</i>	В галузі домінує одна фірма	Надання конкурентоспроможних послуг
<i>Національний рівень конкурентної боротьби</i>	Конкуренція фірм на рівні держави	Реклама
<i>Внутрішньогалузева ознака</i>	В галузі домінує декілька конкуруючих фірм	Конкурентоспроможні. послуг
<i>Національний рівень конкурентної боротьби</i>	Постачальники послуг працюють в межах однієї галузі	Надання схожих за видом послуг
<i>Товарно-видова конкуренція</i>	Конкуренція між послугами одного виду та призначення	Вдосконалення послуг
<i>Нецінові переваги</i>	Переваги відносно результату послуг	Оптимізація витрат та підвищення лояльності до замовника послуг
<i>Марочна інтенсивність</i>	Вказує, яке підприємство відповідальне за готовий продукт	Створення власної марки

Після аналізу конкуренції, проведемо аналіз умов конкуренції у галузі та наведемо результати у таблиці 4.8 нижче. З проведеного аналіз щодо конкуренції на ринку нами було виявлено, що існує можливість виходу на ринок. Як стратегію на початку роботи можна обрати напрямок роботи на встановлення зв'язків з існуючими замовниками послуг.



Таблиця 4.8 – Аналіз конкуренції у галузі

<i>Складові аналізу</i>	<b>Конкуренти</b>		<b>Постачальники</b>	<b>Клієнти</b>	<b>Товари замітники</b>
	<b>Прямі</b>	<b>Потенційні</b>			
	<i>Moodle</i>	Невідомі			
			Значення розміру поставок	Розмір закупівель послуг	Загроз не існує
<i>Висновки</i>	Існує	Дані відсутні	Умови не диктуються постачальниками послуг	Пред'являють вимоги до ціни	Відсутні обмеження для роботи на ринку

З аналізу конкуренції можна визначити тип конкуренції – монополістична. Це відбулося тому, що вони були першими на ринку.

Сформуємо перелік факторів конкурентоспроможності (таблиця 4.9). Поки проект не є реалізованим, ми можемо навести лише приблизні можливі фактори.

Таблиця 4.9 – Обґрунтування факторів конкурентоспроможності

<b>Фактор конкурентоспроможності</b>	<b>Обґрунтування</b>
<i>Точність</i>	Використання сучасних технологій для отримання найбільш точних даних
<i>Додатковий функціонал</i>	Функція визначення «слабких місць» та створення індивідуального плану навчання
<i>Оцінювання студентів</i>	Автоматизований процес оцінювання студентів

На основі вже визначених факторів конкурентоспроможності, проведемо аналіз сильних та слабких сторін стартап-проекту та наведемо результати у таблиці 4.10 нижче.

Таблиця 4.10 – Порівняльний аналіз сильних та слабких сторін проекту

Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів конкурентів у порівнянні з КСК						
		-3	-2	-1	0	+1	+2	+3
<i>Точність</i>	15				+			
<i>Додатковий функціонал</i>	19							+
<i>Тестове середовище</i>	20							+

Фіналізуємо ринковий аналіз можливостей впровадження проекту складанням матриці аналізу сильних та слабких сторін підприємства і його можливостей та загроз у матриці SWOT-аналізу (таблиця 4.11).

Таблиця 4.11 – Матриця SWOT- аналізу стартап-проекту

Сильні сторони	Слабкі сторони
<ul style="list-style-type: none"> <li>– Навчальна платформа для студентів і викладачів</li> <li>– Формування індивідуального плану тестування</li> <li>– Збір статистики для користувачів</li> <li>– Відбір матеріалу для повторення</li> <li>– Оцінювання студентів</li> </ul>	<ul style="list-style-type: none"> <li>– Необхідний вагомий стартовий капітал для початку роботи</li> </ul>
Можливості	Загрози
<ul style="list-style-type: none"> <li>– Створення та використання науково-технічних надбань</li> <li>– Можливість встановлення монополії</li> <li>– Робота з закордонними замовниками</li> </ul>	<ul style="list-style-type: none"> <li>– Кадри</li> <li>– Відмова у співпраці</li> <li>– Небажання користувачів платити за додатковий матеріал</li> <li>– Недостатнє фінансування</li> </ul>

Після проведеного SWOT-аналізу можна зробити висновок, що проект буде конкурентоспроможним за рахунок своїх сильних сторін та можливостей.

Базуючись на SWOT-аналізі, логічним є розробка поведінки виведення стартап-проекту на ринок. Для цього аналізується ймовірність отримання ресурсів за певні строки (таблиця 4.12 нижче).

Таблиця 4.12 – Альтернативи ринкового впровадження стартап-проекту

<b>Альтернатива</b>	<b>Ймовірність отримання ресурсів</b>	<b>Строки реалізації</b>
<i>Підготовка та впровадження програмної частини</i>	Висока	9-12 місяців

Не маючи альтернатив, необхідно дотримуватися одного шляху, а саме – почати підготовку та впровадження програмної частини.

#### 4.5 Розроблення ринкової стратегії проекту

Для розробки ринкової стратегії, визначимо стратегію охоплення ринку через опис гру потенційних споживачів (таблиця 4.13).

Таблиця 4.13 – Вибір цільових груп потенційних споживачів

<b>Опис профілю цільової групи</b>	<b>Готовність споживачів сприйняти послугу</b>	<b>Орієнтовний попит в межах цільової групи</b>	<b>Інтенсивність конкуренції в сегменті</b>	<b>Простота входу в сегмент</b>
<i>Державні навчальні заклади</i>	Готові	Невідомий	Невідомий	Середня
<i>Приватні навчальні заклади</i>	Готові	Невідомий	Невідомий	Середня

Цільовими групами обрано державні та приватні навчальні заклади, бо обидва типа підприємств мають схожі потреби. Для цих сегментів будемо пропонувати систему дистанційного навчання з використанням методу інтервальних повторень та будемо використовувати диференційовану стратегію охоплення ринку. Працюючи із конкретним сегментом, ми будемо розробляти план впливу під конкретні цілі.

З метою роботи в обраному ринковому сегменті, сформуємо базову стратегію розвитку (таблиця 4.14 нижче).

Таблиця 4.14 – Визначення базової стратегії розвитку

<b>Обрана альтернатива розвитку проекту</b>	<b>Стратегія охоплення ринку</b>	<b>Ключові конкурентоспроможні позиції відповідно до обраної альтернативи</b>	<b>Базова стратегія розвитку</b>
Стратегія спеціалізації	Налагодження зв'язків з клієнтами, індивідуальна модифікація ПЗ під потреби	Висока якість та точність, ухил на довготривалі стосунки	Стратегія диференціації

Базовою стратегією оберемо стратегію диференціації – орієнтування на потреби користувача. Альтернативною до неї (у разі провалу) буде обрано стратегію спеціалізації – налаштування під окремий цільовий сегмент.

Далі оберемо стратегію конкурентної поведінки на ринку (таблиця 4.15).

Таблиця 4.15 – Визначення базової стратегії конкурентної поведінки

<b>Чи є проект «першопрохідцем» на ринку?</b>	<b>Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?</b>	<b>Чи буде компанія копіювати основні характеристики товару конкурента, і які?</b>	<b>Стратегія конкурентної поведінки</b>
Проект не є «першопрохідцем»	Пошук нових клієнтів та перехоплення існуючих	Копіювання лише спільного функціоналу та його розширення	Стратегія заняття конкурентної ніші

Обраною є стратегія заняття конкурентної ніші з орієнтацією на декілька сегментів. Головною діяльністю є підтримку клієнтів та розширення функціоналу.

Означимо стратегію позиціонування продукту (таблиця 4.16 нижче).

Підсумуємо обрані стратегії поведінки:

- базова стратегія розвитку – диференціація;
- альтернативна стратегія розвитку – спеціалізація;

- **конкурентна поведінка** – *заняття конкурентної ніші;*
- **обрані сегменти ринку** – *державні і приватні підприємства.*

Таблиця 4.16 – Визначення стратегії позиціонування

<b>Вимоги до товару цільової аудиторії</b>	<b>Базова стратегія розвитку</b>	<b>Ключові конкурентоспроможні позиції власного стартап-проекту</b>	<b>Вибір асоціацій, які мають сформувати комплексну позицію власного проекту</b>
1. Стабільна робота 2. Можливість одночасного підключення багатьох користувачів 3. Наявність підсистеми оцінювання 4. Надання даних щодо успішності студентів	Стратегія диференціації	Висока якість та стабільність, ухил на довготривалі стосунки	1. Стабільність; 2. Надійність; 3. Клієнто-орієнтованість

#### 4.6 Розроблення маркетингової програми стартап-проекту

Планується, що кінцевий споживач після підписання контракту на закупівлю продукції та послуг, отримає доступ до системи дистанційного навчання з використанням методики, заснованої на методі інтервальних повторень.

На першому кроці, сформуємо маркетингову концепцію товару, який отримає кінцевий споживач.

Для цього підсумуємо результати попереднього аналізу конкурентоспроможності товару (таблиця 4.17 нижче).

Таблиця 4.17 – Визначення ключових переваг концепцій потенційного товару (послуги)

<b>Потреба</b>	<b>Вигода, яку пропонує товар (послуга)</b>	<b>Ключові переваги перед конкурентами</b>
<i>Комфорт використання системи дистанційного навчання</i>	Надання алгоритму в системі, які буде визначати «слабкі місця» для студентів з необхідністю частішого повторення, а також формувати індивідуальний план для тестувань такого виду	Наявність алгоритму інтервального навчання з урахуванням індивідуальних особливостей студентів
<i>Оптимізація витрат на обслуговування СДН</i>		

Робимо висновок, що послуга має переваги над конкурентами та відповідає вимогам споживачів.

Розробимо трирівневу маркетингову модель товару, де уточнимо ідею послуги та особливості її надання (таблиця 4.18 нижче). Наступним кроком буде визначення цінових меж для потенційного товару на основі даних про виробництво та товари аналоги.

Ми не в змозі оцінити вартість програмного забезпечення конкурента, але можемо зробити певні припущення щодо апаратного забезпечення, яке використовується для власного хостингу. Зведемо ці данні у таблиці 4.19 нижче.

Правильним ходом буде встановлення цін, нижчим за ціни конкурентів. Товарів замінників виявлено не було, а вартість (часткова, лише за апаратну частину) товарів аналогів досить висока. Тому необхідно намагатися залишатися у межах цінового діапазону близько 15000-20000 гривень за одиницю клієнтського апаратного забезпечення разом із встановленим програмним забезпеченням і підтримкою на рік.

Таблиця 4.18 – Опис трьох рівнів моделі товару

Рівні послуг	Сутність та складові		
I. Послуга за задумом	Індивідуальний план навчання на основі методу інтервальних повторень		
II. Послуга у Реальному виконанні	Властивості/характеристики	<i>М/Нм</i>	<i>Вр/Тх /Тл/Е/Ор</i>
	<i>Якість</i>	Нм	Тл
	<i>Точність</i>	Нм	Тл
	<i>Імплементация у навчальному закладі</i>	М	Тх
	<i>Ціна</i>	Нм	Е
	<i>Підтримка</i>	М	Тх
	Якість: тестування програмного забезпечення за сценаріями використання		
	Постачання: друкований примірник інструкції з експлуатації з особливим серійним номером та гарантійним талоном.		
Марка: назва системи та унікальний номер для навчального закладу			
III. Послуга із підкріпленням	Акція при закупці продукції оптом та при користуванні на протязі певного довгого періоду		
За рахунок чого потенційний товар буде захищено від копіювання: захист інтелектуальної власності, патент на винахід.			

Таблиця 4.19 – Визначення меж встановлення ціни

Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
Відсутні товари замінники	(Аналог хмарного хостингу)  18000-25000 гривень	Середній	(клієнтської апаратної частини)  15000-2000 гривень

Оберемо систему збуту, яка буде відповідати наступним вимогам (таблиця 4.20 нижче).

Таблиця 4.20 – Формування системи збуту

<b>Специфіка закупівельної поведінки цільових клієнтів</b>	<b>Функції збуту, які має виконувати постачальник товару</b>	<b>Глибина каналу збуту</b>	<b>Оптимальна система збуту</b>
<i>Звична купівля (програмного забезпечення)</i>	Постачальник виконує такі основні функції: <ul style="list-style-type: none"> <li>– розробка;</li> <li>– вдосконалення;</li> <li>– підтримка;</li> <li>– продаж.</li> </ul>	Канал першого рівня	Пряма система збуту з пошуком клієнтів

Розробимо концепцію маркетингової комунікації, за основу якої візьмемо стратегії позиціонування та специфіку поведінки клієнта (таблиця 4.21).

Таблиця 4.21 – Концепція маркетингових комунікацій

<b>Специфіка поведінки цільових клієнтів</b>	<b>Канали комунікації</b>	<b>Ключові позиції для позиціонування</b>	<b>Концепція рекламного звернення</b>
<i>Пошук шляхів мінімізації витрат</i>	Інтернет ресурси; Спеціалізовані виставки;	Канал першого рівня	Зменшення витрат на обслуговування маршрутів та підвищення лояльності пасажирів

В результаті проведення досліджень, у цьому розділі було визначено ключові переваги над товарами та послугами конкурентів, виявлено канали збуту та комунікації з клієнтами а також приблизно оцінено кошовність програмної клієнтської частини системи.



### **Висновок до розділу**

Мета цього розділу роботи полягала в тому, щоб сформувати підприємницьку ідею у вигляді стартап-проекту. У такому проекті першочерговою дією виступив пошук вже існуючих аналогів продукту або послуги, яку планується вивести на ринок. Під час аналізу було виявлено декілька закордонних системи, що мають схожий функціонал. Проте неможливо назвати їх прямими конкурентами, але вони за своєю сутністю можуть внести частину впливу на формування кінцевого продукту.

Наступним кроком була оцінка спроможності втілення пропонованої ідеї у життя з технічної позиції. Для цього ідею поділено на частини, і з кожної було виокремлено технічну частину, а також було оцінено її доступність. З проведених досліджень стало ясно, що для реалізації системи не існує жодних перешкод.

Стартап належить до переліку найбільш ризикованих видів підприємницької діяльності. Саме тому, для зменшення негативного впливу факторів різноманітного типу на розвиток підприємницької діяльності, було проведено аналіз загроз та пошук шляхів їх обходу або, принаймні, мінімізації частини впливу.

Останнім кроком стала побудова маркетингового плану, яка надасть можливість реалізувати проект і вивести стартап на ринок.

У результаті було досягнуто наступне: для системи дистанційного навчання створено стартап-проект, який регулює її введення на український ринок.

## ЗАГАЛЬНІ ВИСНОВКИ

У даній роботі розглядалась система підтримки методики навчального процесу, заснованої на методі інтервальних повторень. У ході роботи були покриті всі важливі аспекти, а саме – проектні рішення, математичне та технологічне забезпечення, опис стартап-проекту.

Послідовно, крок за кроком, були пройдені всі етапи, від ідеї до підготовки системи до реалізації. Було переглянуто дуже багато інформаційних джерел, проведено ознайомлення з аналогами та можливими конкурентами, визначені їхні сильні та слабкі сторони. Завдяки цьому вдалося визначити бізнес-логіку для системи, що розроблялась.

У математичному забезпеченні була складена задача пошуку оптимального терміну тестування, яка заснована на поєднанні методу інтервальних повторень та кривої забування німецького вченого Еббінгауза. Були розглянуті схожі методи, був запропоновано власний метод, який було покладено в алгоритм, що буде використовуватися в системі для урахування особистісних навичок при формуванні індивідуального плану навчання.

В технологічному розділі були розглянуті засоби розробки, обрана та описана архітектура програмного забезпечення, наведена інструкція для користувачів та описані проведені випробування.

В останньому розділі була описана стартап-ідея проекту. Оскільки ідея містить елементи нововведення, то був проведений детальний аналіз ринку, наявних конкурентів, факторів загроз тощо – все, що стосується виведення системи на ринок.

В результаті виконаної роботи наявна система, готова до попереднього запуску в режимі публічного тестування для подальшої розробки та її імплементації в навчальний процес. Це є головним досягненням даної роботи.

## ПЕРЕЛІК ПОСИЛАНЬ

1. Романчук І.О. Система підтримки методики навчального процесу, заснованої на методі інтервальних повторень / І.О. Романчук, В.Д. Попенко // Матеріали III всеукраїнської науково-практичної конференції молодих вчених та студентів «Інформаційні системи та технології управління» (ІСТУ-2019) – м. Київ.: НТУУ «КПІ ім. Ігоря Сікорського», 20-22 листопада 2019 р. – С. 210-215.
2. Романчук І.О. Вдосконалення методики навчального процесу з використанням методу інтервальних повторень / І.О. Романчук, В.Д. Попенко // «Наукові записки. Серія: Педагогічні науки» – вип. 183, ЦДПУ ім. Володимира Винниченка, 2020 – 16 с.
3. Codecademy: Learn to Code - for Free [Електронний ресурс] // Режим доступу: <https://www.codecademy.com>
4. Duolingo [Електронний ресурс] // Режим доступу: <https://duolingo.com/>
5. How we learn how you learn | Blog – Duolingo [Електронний ресурс] // Режим доступу: <https://making.duolingo.com/how-we-learn-how-you-learn>
6. "Human Memory: Theory and Practice", Alan D. Baddeley, 1997
7. Б. Демида, С. Сагайдак, І. Копил «Системи дистанційного навчання: огляд, аналіз, вибір» / Національний університет “Львівська політехніка”, кафедра автоматизованих систем управління [Електронний ресурс] // Режим доступу: <http://ena.lp.edu.ua/bitstream/ntb/10662/1/14.pdf>
8. ATutor User Documentation [Електронний ресурс] // Режим доступу: <http://help.atutor.ca/general/>
9. Claroline – Documentation [Електронний ресурс] // Режим доступу: [http://doc.claroline.net/en/index.php/Main\\_Page](http://doc.claroline.net/en/index.php/Main_Page)
10. eFront Руководство преподавателя / ООО “АББРИС”
11. Белозубов А.В., Николаев Д.Г. Система дистанционного обучения Moodle: учебно-методическое пособие. – СПб., 2007.
12. Андреев А.В., Андреева С.В, Доценко И.Б. Практика электронного обучения с использованием Moodle. –Таганрог: ТТИ ЮФУ, 2008. – 146 с
13. Система дистанционного обучения SharePointLMS, Belitsoft

14. Moodle [Электронный ресурс] // Режим доступа: <https://moodle.org/?lang=uk>
15. Л. А. Карпенко, М. Г. Ярошевский. История психологии в лицах. Персоналии / Под общ. ред. А. В. Петровского. — М.: ПЕР СЭ, 2005. — Т. 6. — 784 с. — (Психологический лексикон. Энциклопедический словарь в шести томах). — 3000 экз. — ISBN 5-9292-0136-6
16. Spitzer, H. F. (1939). Studies in retention. *Journal of Educational Psychology*, 30, 641—657
17. Landauer, T. K., & Bjork, R. A. (1978). Optimum rehearsal patterns and name learning. In M. Gruneberg, P. E. Morris, & R. N. Sykes (Eds.), *Practical aspects of memory* (pp. 625—632). London: Academic Press
18. Melton, A. W. (1970). The situation with respect to the spacing of repetitions and memory. *Journal of Verbal Learning and Verbal Behavior*, 9, 596—606
19. Pimsleur, Paul (1967), "A Memory Schedule", *The Modern Language Journal* (The Modern Language Journal, Vol. 51, No. 2) . — Т. 51 (2): 73–75
20. Gradint, a program for self-study of foreign languages [Электронный ресурс] // Режим доступа: <http://ssb22.user.srcf.net/gradint/>
21. Learning tools and flashcards - for free! | Quizlet [Электронный ресурс] // Режим доступа: <https://quizlet.com/ru>
22. StudyBlue: Online Flashcards, Homework Help & Textbook Solutions [Электронный ресурс] // Режим доступа: <https://www.studyblue.com/>
23. StudyStack | Flashcards and Study Games [Электронный ресурс] // Режим доступа: <https://www.studystack.com/>
24. Геймификация [Электронный ресурс] // Режим доступа: <https://4brain.ru/gamification/>
25. Brainscape - Smart Flashcards on the App Store [Электронный ресурс] // Режим доступа: <https://apps.apple.com/us/app/brainscape-smart-flashcards/id442415567>
26. Anki - powerful, intelligent flashcards [Электронный ресурс] // Режим доступа: <https://apps.ankiweb.net/>
27. A Trainable Spaced Repetition Model for Language Learning, Burr Settles, Brendan Meeder, *Proceedings of the 54th Annual Meeting of the Association for*

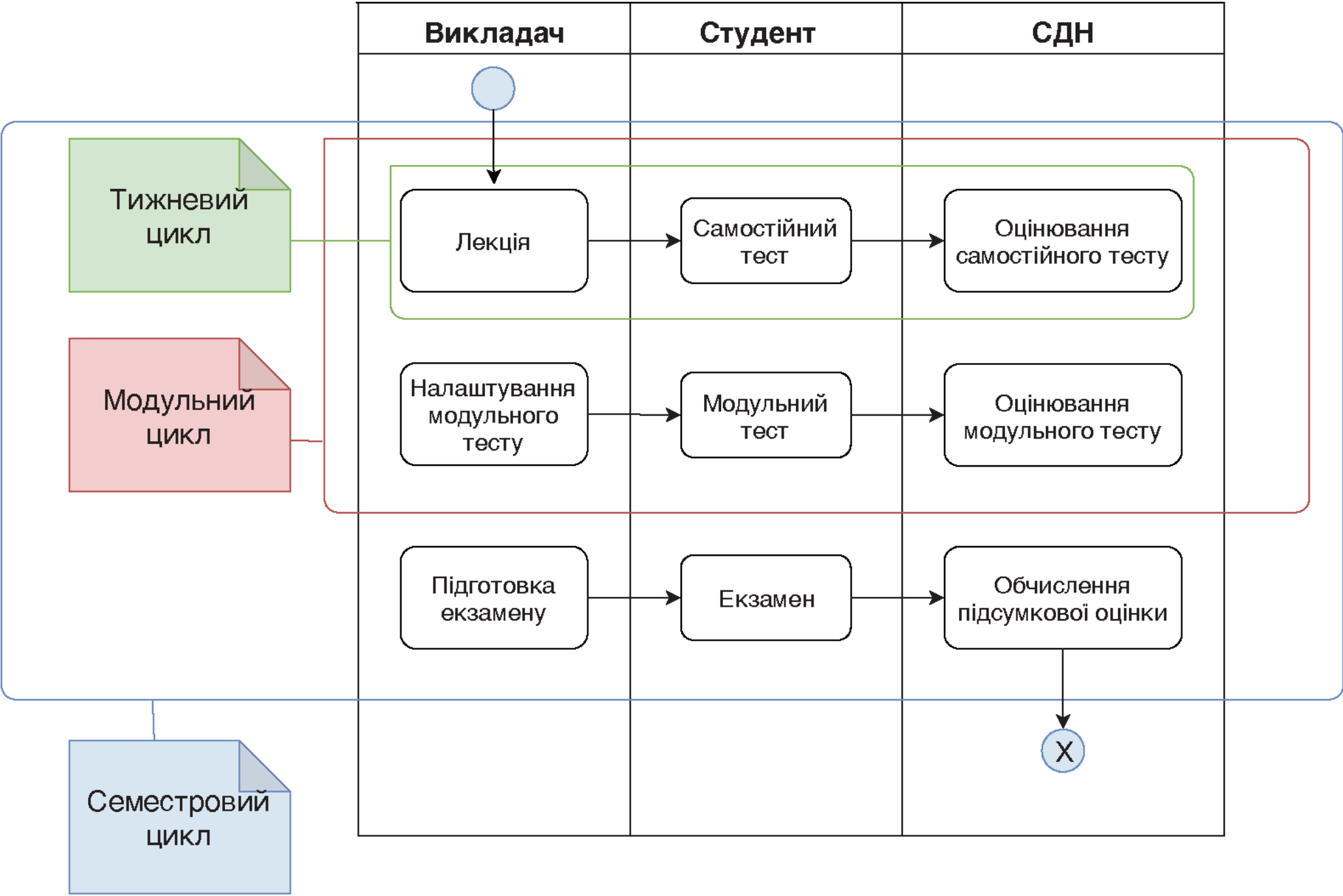
- Computational Linguistics (Volume 1: Long Papers), August 2016, pages 1848-1858, DOI 10.18653/v1/P16-1174
28. Caple, C. (1996). "The Effects of Spaced Practice and Spaced Review on Recall and Retention Using Computer Assisted Instruction". Dissertation for the degree of Doctor of Education, North Carolina State University.
  29. Мобильные и веб-приложения: что является лучшим решением? [Электронный ресурс] // Режим доступа: <https://smartum.pro/ru/blog-ru/razrabotka-mobilnykh-i-web-prilozheniy/>
  30. Как компании могут извлечь выгоду от внедрения мобильного приложения [Электронный ресурс] // Режим доступа: <https://smartum.pro/ru/blog-ru/kak-razrabotka-mobilnogo-prilozheniya-sposobstvuyet-uspekhy-kompanii/>
  31. What Is SEO / Search Engine Optimization? - Search Engine Land [Электронный ресурс] // Режим доступа: <https://searchengineland.com/guide/what-is-seo>
  32. Создаем простое веб приложение при помощи Node.js [Электронный ресурс] // Режим доступа: <https://umbrellait.com/ru/how-to-build-a-simple-web-application-using-node-js/>
  33. Сравнение JavaScript фреймворков Vue.js, React и Angular (2019) | статьи о программировании mkdev [Электронный ресурс] // Режим доступа: <https://mkdev.me/posts/sravnenie-javascript-freymvorkov-vue-js-react-i-angular-2019>
  34. Angular React Vue.js Jobs, Employment | Indeed.com [Электронный ресурс] // Режим доступа: <https://www.indeed.com/q-Angular-React-vue.js-jobs.html>
  35. Stack Overflow Developer Survey 2017 [Электронный ресурс] // Режим доступа: <https://insights.stackoverflow.com/survey/2017>
  36. vuejs/vue: Vue.js is a progressive, incrementally-adoptable JavaScript framework for building UI on the web [Электронный ресурс] // Режим доступа: <https://github.com/vuejs/vue>
  37. WebStorm [Электронный ресурс] // Режим доступа: <https://jetbrains.ru/products/webstorm/>

38. IDE - Словник | MDN [Електронний ресурс] // Режим доступу:  
<https://developer.mozilla.org/uk/docs/Glossary/IDE>
39. IntelliJ IDEA: IDE для професійної розробки на Java від JetBrains  
[Електронний ресурс] // Режим доступу: <https://www.jetbrains.com/ru-ru/idea/>
40. Firebase [Електронний ресурс] // Режим доступу: <https://firebase.google.com>
41. Використовуйте можливості Firebase для розробки | Android Developers  
[Електронний ресурс] // Режим доступу:
42. Technomatix | Moodle [Електронний ресурс] // Режим доступу:  
<https://moodle.com/partners/technomatix/>
43. MoodleCloud: Free Hosting Services from the makers of Moodle [Електронний  
ресурс] // Режим доступу: <https://moodlecloud.com/app/en/>
44. Moodle statistics [Електронний ресурс] // Режим доступу: <https://stats.moodle.org>
45. В. М. Томашевський, Ю.Л. Новіков, П.А. Камінська / Наукові праці: Науково-  
методичний журнал. Т. 160. Вип. 122. Комп'ютерні технології. Миколаїв: Вид.  
ЧДУ ім. Петра Могили. – 2011. С. 146-157
46. Погребнюк І.М. Моделювання сценаріїв адаптивного навчання з використання  
м мереж Петрі / І.М. Погребнюк, В.М. Томашевський / Вісник НТУУ «КПІ». І  
нформатика, управління та обчислювальна техніка: Зб. наук. пр. – К.: Век+, –  
2012. – № 55. – С. 38-45
47. Новіков Ю.Л. інформаційна технологія створення дистанційних інтернет  
систем навчання [Електронний ресурс] // Режим доступу:  
<http://compi.com.ua/novikov-yurij-leonidovich-informacijna-tehnologiya-stvorennnya.html>

## **ДОДАТОК А**

### **Графічний матеріал**

Плакати 1. Схематична діяльність до оптимізації



Демонстраційний плакат до магістерської дисертації  
на тему «Система підтримки методики навчального процесу, заснованої на методі інтервальних повторень»

Виконав студент гр. ІС-81мп

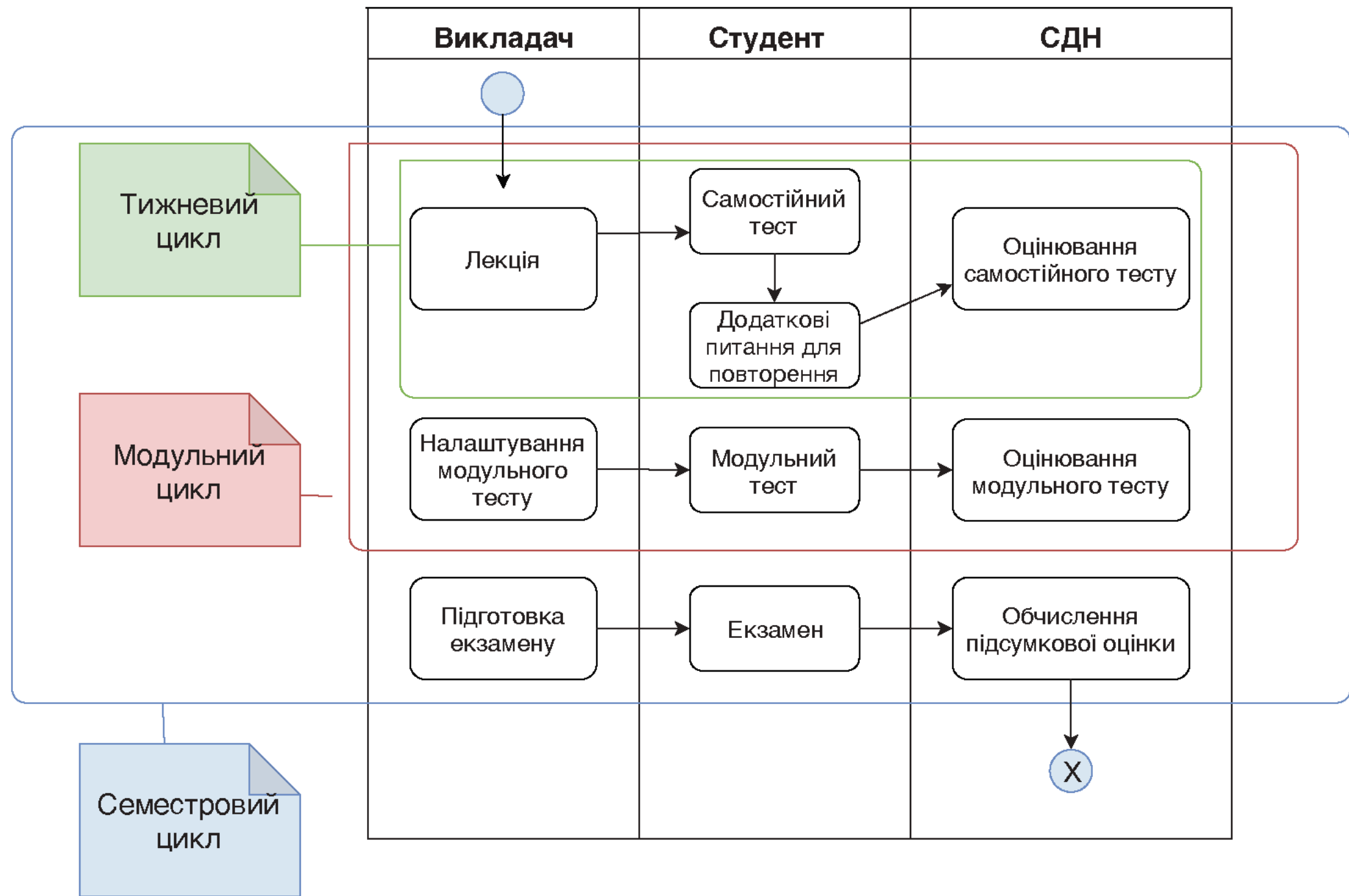
Романчук І.О.

Керівник

Попенко В.Д.



# Плакат 2. Схема структурна діяльності після оптимізації

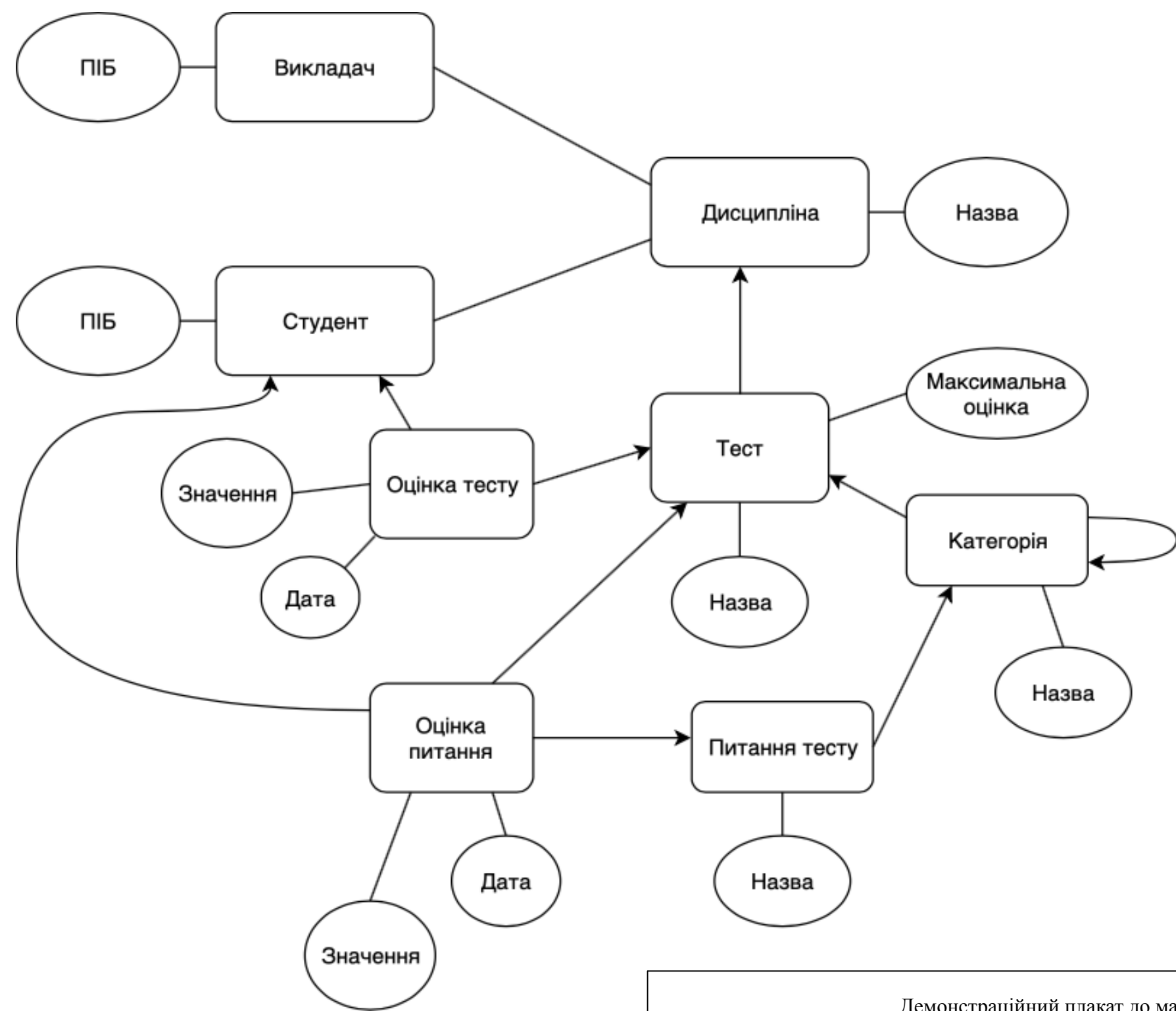


Демонстраційний плакат до магістерської дисертації  
на тему «Система підтримки методики навчального процесу, заснованої на методі інтервальних повторень»

Виконав студент гр. ІС-81мп  
Керівник

Романчук І.О.  
Попенко В.Д.

# Плакат 3. Модель "сутність-зв'язок"



Демонстраційний плакат до магістерської дисертації  
на тему «Система підтримки методики навчального процесу, заснованої на методі інтервальних повторень»

Виконав студент гр. ІС-81мп  
Керівник

Романчук І.О.  
Попенко В.Д.

## Плакат 4. Крива забування в разі невдалого проходження тесту



Демонстраційний плакат до магістерської дисертації  
на тему «Система підтримки методики навчального процесу, заснованої на методі інтервальних повторень»

Виконав студент гр. ІС-81мп

Романчук І.О.

Керівник

Попенко В.Д.

## Плакат 5. Крива забування в разі вдалого проходження тесту



Демонстраційний плакат до магістерської дисертації  
на тему «Система підтримки методики навчального процесу, заснованої на методі інтервальних повторень»

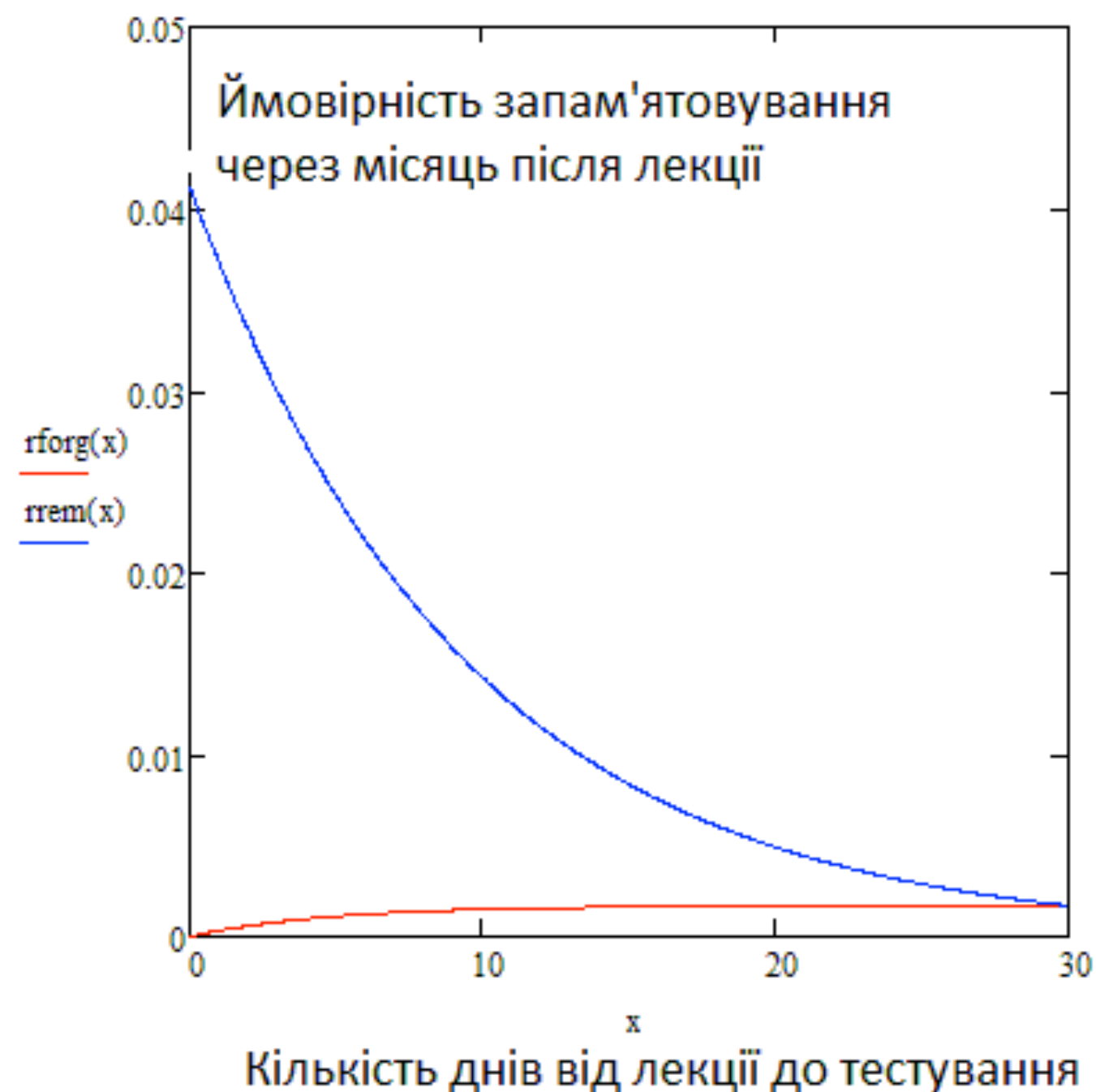
Виконав студент гр. ІС-81мп

Романчук І.О.

Керівник

Попенко В.Д.

## Плакат 6. Ймовірність запам'ятовування за місяць, внески випадків невдалого (червона крива) і вдалого (синя крива) тестування



Демонстраційний плакат до магістерської дисертації  
на тему «Система підтримки методики навчального процесу, заснованої на методі інтервальних повторень»

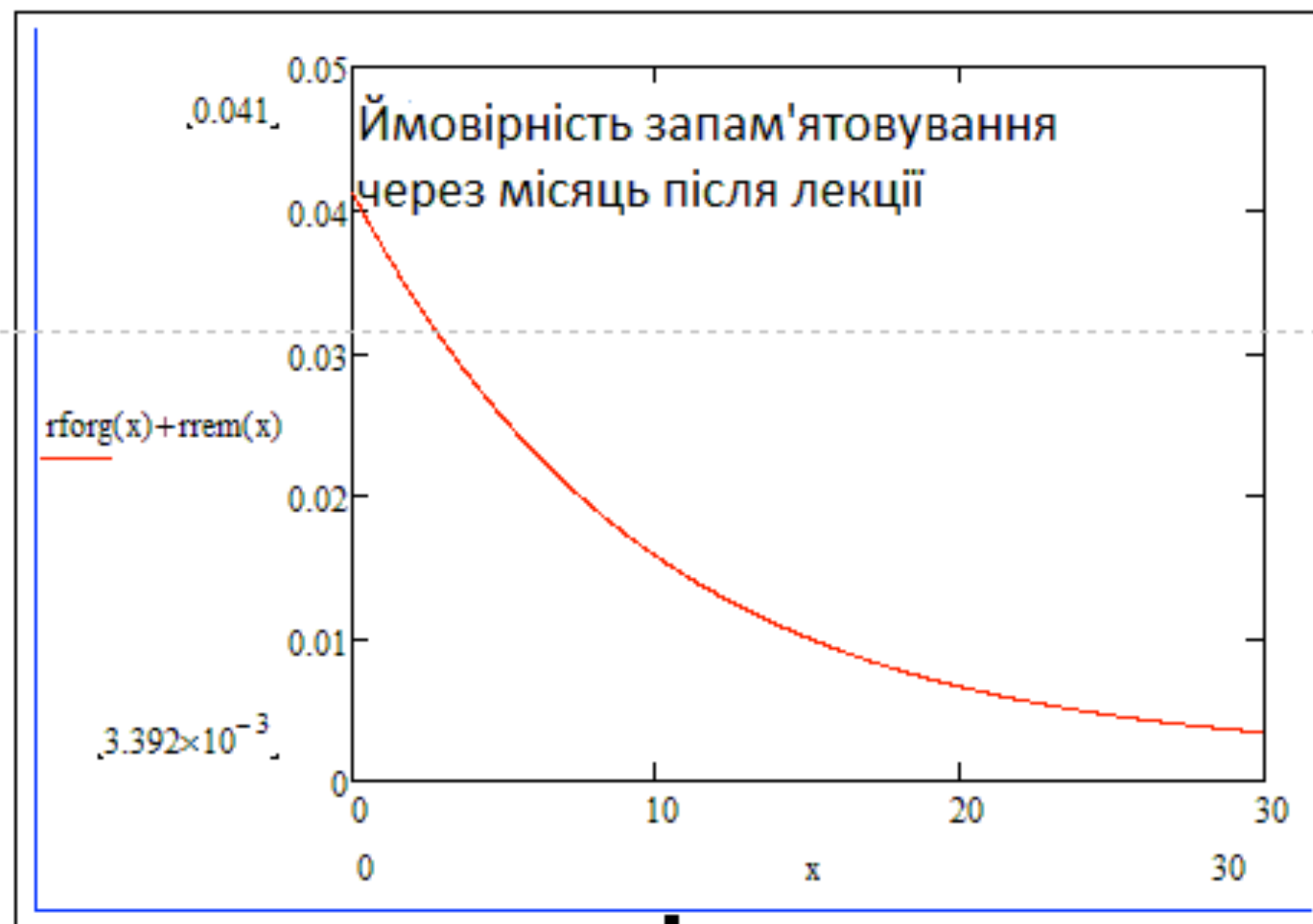
Виконав студент гр. ІС-81мп

Романчук І.О.

Керівник

Попенко В.Д.

## Плакат 7. Ймовірність запам'ятовування за місяць, сумарний графік



Кількість днів від лекції до тестування

Демонстраційний плакат до магістерської дисертації  
на тему «Система підтримки методики навчального процесу, заснованої на методі інтервальних повторень»

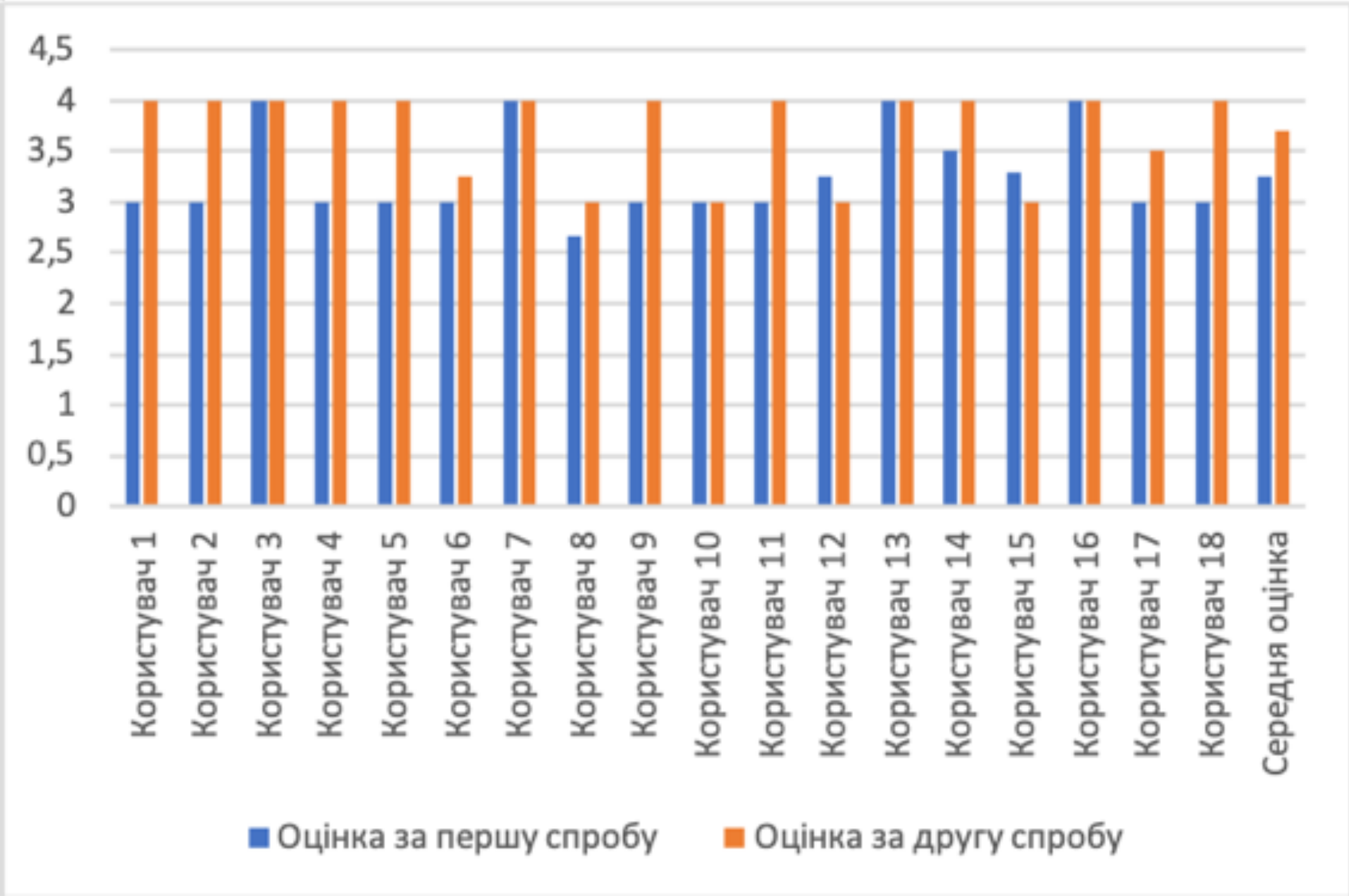
Виконав студент гр. ІС-81мп

Романчук І.О.

Керівник

Попенко В.Д.

# Плакат 8. Результати проведеного експерименту з використанням методу інтервальних повторень



Демонстраційний плакат до магістерської дисертації  
на тему «Система підтримки методики навчального процесу, заснованої на методі інтервальних повторень»

Виконав студент гр. ІС-81мп

Романчук І.О.

Керівник

Попенко В.Д.